

KONINKLIJK NEDERLANDS  
METEOROLOGISCH INSTITUUT

D e B i l t

WETENSCHAPPELIJK RAPPORT

W.R. 70-3

Th. de Crook

Het berekenen van stroomlijnen  
uit een discreet divergent vectorveld

De Bilt, 1970

Publikatienummer: K.N.M.I. W.R. 70-3 (I)

All rights reserved

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van het K.N.M.I., De Bilt.

U.D.C. 551.465.536  
.....  
.....

## I N H O U D

Summary

1. Inleiding

2. Probleemstelling

3. De methode van Pritchard

4. De methode van Heyna

5. De methode van Hawkins en Rosenthal

6. De methode van Picard

6.1. De oplossingstechniek

6.2. Uitbreiding van de methode

6.3. Foutenbeschouwing

6.4. Toelichting bij het stroomdiagram

6.5. Het programma SPM

6.6 Toepassingen van het progr. SPM, waarvan de analytische oplossing bekend is

6.7. Toepassingen uit de praktijk

6.8. Conclusie

7. Literatuur

BIJLAGE 1 (het stroomdiagram)

BIJLAGE 2 (het Algol programma)

FIGUREN

Summary

The title of this paper is: "Computation of streamlines from a discrete divergent vector field".

This investigation has been started to develop an objective method to compute and to plot streamlines with a computer.

For divergent vector fields it is difficult to compute streamlines with a computer. For non-divergent or quasi-non-divergent vector fields this is much easier. The methods which exist, give only useful results for vector fields with small divergence, except the method of Heyna, which is however less suitable for the computer.

In section 3, 4 and 5 three existing methods for the computation of streamlines are discussed briefly, respectively the methods of Pritchard, Heyna and Hawkins-Rosenthal.

The method of Pritchard is only useful for non-divergent or quasi-non-divergent vector fields. The method of Hawkins and Rosenthal fails when the divergence is too large.

The developed method is described in section 6 and is applicable to vector fields with divergence. It is an extension of the method of Picard. For this method it is necessary that the vectors are known at the gridpoints of a rectangular network. The new method is applied to a mean windfield of the North-Atlantic Ocean. The results of these applications are at least of the same quality as the results, obtained on a subjective way.

## 1. Inleiding

Voor het bepalen van stroomlijnen uit een gegeven vectorveld zijn er subjectieve en objectieve methoden. In dit verslag beperken wij ons tot objectieve methoden, daar deze zich i.h.a. goed lenen voor berekeningen met een computer en zodoende snel een oplossing verkregen kan worden.

Indien er divergentie optreedt, is het objectief bepalen van stroomlijnen een moeilijke zaak. De thans bestaande methoden geven slechts bruikbare resultaten als de divergentie klein is, met uitzondering van de methode van Heyna (1), die echter minder geschikt is voor verwerking met de computer. Voor divergentievrije of quasi-divergentievrije velden is het berekenen van stroomlijnen aanzienlijk gemakkelijker.

In de punten 3, 4 en 5 worden achtereenvolgens in het kort de methode van Pritchard (2), de methode van Heyna (1) en de methode van Hawkins en Rosenthal (3) besproken.

De methode van Pritchard is uitsluitend geschikt voor divergentievrije of quasi-divergentievrije velden. De methode van Hawkins en Rosenthal biedt ruimere mogelijkheden, doch faalt indien de divergentie te groot wordt.

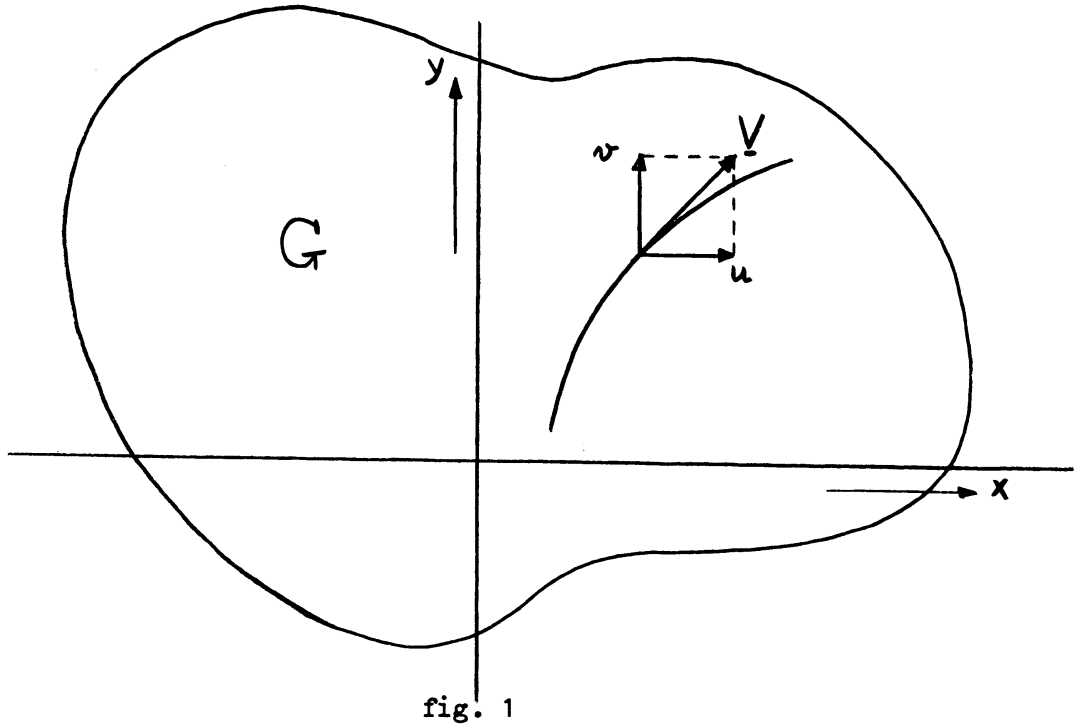
In punt 6 wordt de methode van Picard uitvoerig besproken. Deze methode is uitgebreid en aangepast voor verwerking met de computer en kan het verloop van de stroomlijnen bepalen, ook voor niet divergentievrije velden. De afstand van de stroomlijnen die een indicatie moet geven van de grootte van de snelheid is een moeilijk punt, dat nog niet is opgelost.

Het algolprogramma SPM (Het berekenen van Stroomlijnen door gegeven beginpunten volgens de Picard-Methode) is geschreven voor de EL-X8; de invoergegevens staan op ponsband; de resultaten worden op ponsband geponst en kunnen tevens met behulp van de plotter worden getekend.

De tijdsduur van het programma SPM is afhankelijk van het aantal te berekenen stroomlijnen, het aantal stappen per stroomlijn en het aantal iteraties per stap.

## 2. Probleemstelling

In een gebied  $G$  is een vectorveld  $V$  gegeven door zijn componenten  $u$  en  $v$ , waarbij  $u$  de  $x$ -component en  $v$  de  $y$ -component voorstelt ( $u$  en  $v$  zijn beide functies van  $x$  en  $y$ ).



Onder een stroomlijn wordt verstaan een kromme in het  $x, y$ -vlak, waarvan de raaklijn in elk punt samenvalt met de richting van de vector  $\underline{V}$  ter plaatse. Een stroomlijn is dus een oplossing van de differentiaalvergelijking van de eerste orde:

$$\frac{dy}{dx} = \frac{v(x,y)}{u(x,y)} \quad (2.1)$$

De oplossing van de diff. vgl. (2.1.) kan worden voorgesteld door

$$\psi(x,y) = \text{constant} \quad (2.2)$$

Het vectorveld  $\underline{V}$  in het gebied G kan analytisch gegeven zijn, d.w.z.  $u(x,y)$  en  $v(x,y)$  zijn als analytische uitdrukkingen gedefinieerd.

Het is echter ook mogelijk en dat is in de praktijk meestal het geval, dat  $u$  en  $v$  alleen bekend zijn in een aantal discrete punten b.v. in de roosterpunten van een rechthoekig rooster.

We beperken ons hier tot het tweede geval, waarbij  $u$  en  $v$  bekend zijn in de roosterpunten van een rechthoekig netwerk.

Hierna volgen een aantal methoden, waarmee we de diff. vgl. (2.1) kunnen oplossen en dus de stroomlijnen kunnen bepalen.

### 3. De methode van Pritchard

Ter oplossing van de differentiaalvergelijking (2.1) behandelen we eerst de methode van Pritchard.

Door volledige differentiatie van de oplossing

$$\psi(x,y) = \text{constante} \quad (3.1)$$

krijgt men

$$\frac{\partial \psi}{\partial x} dx + \frac{\partial \psi}{\partial y} dy = 0 \quad (3.2)$$

De vergelijking (2.1) kan geschreven worden als

$$v(x,y) dx - u(x,y) dy = 0 \quad (3.3)$$

De vergelijkingen (3.2) en (3.3) vormen een stelsel van homogene lineaire vergelijkingen in de differentiaal  $dx$ ,  $dy$  en hebben een oplossing als er evenredigheid tussen de coëfficiënten bestaat. Hieruit volgt de existentie van een functie  $\lambda(x,y)$  zo dat

$$\frac{\partial \psi}{\partial x} = -\lambda(x,y) \cdot v(x,y) \quad (3.4)$$

en

$$\frac{\partial \psi}{\partial y} = \lambda(x,y) \cdot u(x,y) \quad (3.5)$$

De functie  $\lambda(x,y)$  noemt men de "integrerende factor". Differentiëren we (3.4) naar  $y$  en (3.5) naar  $x$  en trekken beide vergelijkingen af, dan vinden we:

$$\frac{\partial \lambda x}{\partial x} + \frac{\partial \lambda v}{\partial y} = 0 \quad (3.6)$$

Nemen we aan dat  $\lambda(x,y)$  constant is, b.v.  $\lambda(x,y) = 1$  dan wordt (3.6):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3.7)$$

d.w.z. dat het vectorveld  $\underline{V}$  divergentievrij is.

Voor het geval het vectorveld divergentievrij is, noemen we de functie  $\psi$  de stroomfunctie.

De stroomfunctie wordt dus beschreven door in de vergelijkingen (3.4) en (3.5)  $\lambda(x,y) = 1$  te substitueren, dus door:

$$\frac{\partial \psi}{\partial x} = -v(x,y) \quad (3.8)$$

en

$$\frac{\partial \psi}{\partial y} = u(x,y) \quad (3.9)$$

De formules (3.8) en (3.9) werden door PRITCHARD gebruikt voor het bepalen van de stroomfunctie, dus voor het berekenen van stroomlijnen in een divergentievrij veld. Ook voor quasi-divergentievrije velden is het mogelijk de stroomlijnen vrij nauwkeurig te berekenen. Onder quasi-divergentievrij dient in dit geval te worden verstaan dat de divergentie klein is over een groot gebied en bovendien vaak van teken wisselt.

Voor meer informatie over deze methode zij verwezen naar (2) en (1).

#### 4. De methode van Heyna

Met de methode van Heyna (1) kunnen stroomlijnen berekend worden in zowel divergentievrije velden als in niet divergentievrije velden. Integenstelling tot de methode van Pritchard wordt hier de integrerende factor  $\lambda(x,y)$  niet constant verondersteld. De integrerende factor wordt bepaald door het oplossen van de differentiaalvergelijking (3.6) nl.:

$$\frac{\partial \lambda u}{\partial x} + \frac{\partial \lambda v}{\partial y} = 0 \quad (4.1)$$

Stellen we  $\hat{u} = \lambda u$  en  $\hat{v} = \lambda v$  dan geldt: (4.2)

$$\frac{\partial \hat{u}}{\partial x} + \frac{\partial \hat{v}}{\partial y} = 0 \quad (4.3)$$

d.w.z. dat het getransformeerde veld  $\hat{V}$ , waarin  $\hat{u}$  de x-component en  $\hat{v}$  de y-component is, divergentievrij is en dus een stroomfunctie heeft.

De stroomfunctie die wordt bepaald door de formules

$$\frac{\partial \psi}{\partial x} = -\hat{v}(x,y) \quad (4.4)$$

en

$$\frac{\partial \psi}{\partial y} = \hat{u}(x,y) \quad (4.5)$$

kunnen we vinden door numerieke integratie.

De vergelijkingen (4.4) en (4.5) zijn verkregen door substitutie van (4.2) in (3.4) en (3.5).

Daar u en v beide met  $\lambda$  worden vermenigvuldigd en de richting van de vector  $\underline{V}$  dus niet verandert (wel de grootte van  $\underline{V}$ ), is een stroomlijn in het getransformeerde veld ook een stroomlijn in het oorspronkelijke veld.

Met het bepalen van de stroomfunctie in het getransformeerde divergentievrije veld, hebben we dus tevens de stroomlijnen in het oorspronkelijke veld gevonden.



In eerste instantie lijkt deze methode zeer eenvoudig, de moeilijkheden liggen echter bij het oplossen van de partiële differentiaalvergelijking (4.1), dus bij het bepalen van de  $\lambda$ 's (in de roosterpunten) met enkele speciale voorzieningen (o.a. het aanbrengen van singuliere lijnen of sneden) is het mogelijk deze moeilijkheden te overwinnen, doch deze maken de methode minder geschikt voor verwerking met een computer (zie (1)).

### 5. De methode van Hawkins en Rosenthal

In deze methode, die uitvoerig wordt behandeld in (3), wordt het vectorveld  $\underline{V}$  (Hawkins en Rosenthal nemen hiervoor een windveld) beschreven door:

$$\underline{V} = \underline{k} \times \nabla\psi + \nabla\chi \quad (5.1)$$

waarbij  $\underline{V}$  de horizontale wind voorstelt,  $\underline{k}$  de verticale eenheidsvector,  $\psi$  de horizontale stroomfunctie en  $\chi$  de horizontale snelheidspotentiaal.

Uit (5.1) kunnen we gemakkelijk afleiden dat:

$$1e: \nabla^2\chi = \nabla \cdot \underline{V} \quad (5.2)$$

$$2e: \nabla^2\psi = \underline{k} \cdot (\nabla \times \underline{V}) \quad (5.3)$$

$$3e: \frac{\partial\psi}{\partial s} = -v_n + \frac{\partial\chi}{\partial n} \quad (5.4)$$

$$4e: \frac{\partial\psi}{\partial n} = v_s - \frac{\partial\chi}{\partial s} \quad (5.5)$$

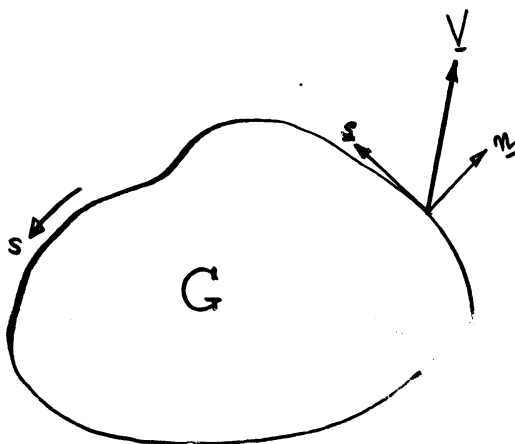


fig. 2

Hierbij is  $\underline{n}$  de eenheidsvector, die loodrecht staat op de rand van het gebied G, waar  $\underline{V}$  bekend is en  $\underline{s}$  de eenheidsvector loodrecht op  $\underline{n}$ . De snelheidscomponenten langs  $\underline{n}$  en  $\underline{s}$  noemen we  $v_n$  en  $v_s$  (zie fig. 2).

Het bepalen van de stroomfunctie  $\psi$  uit het windveld kan op verschillende manieren geschieden, bv. als volgt:

Los de vergelijking  $\nabla^2\chi = \nabla \cdot \underline{V}$  op met  $\chi = 0$  op de rand.

Bepaal hierna  $\frac{\partial\chi}{\partial n}$  op de rand m.b.v. lineaire differenties. Teneinde de randwaarden van  $\psi$  te verkrijgen, integreren we (5.4) langs de rand van het veld (m.b.v. de trapeziumregel).

Tenslotte kunnen we met de berekende randwaarden voor  $\psi$  de Poissonvergelijking (5.3) oplossen.

Deze methode die uitermate geschikt is voor verwerking met een computer heeft het nadeel dat indien de divergentie te groot wordt, de randwaarden van de stroomfunctie  $\psi$  niet nauwkeurig genoeg berekend kunnen worden.

Hier kan wel enige verbetering in worden aangebracht door de methode te verfijnen door b.v.  $\frac{\partial X}{\partial n}$  m.b.v. centrale differenties te bepalen, doch de methode blijft ongeschikt voor sterke divergente velden.

Een ander bezwaar is dat de stroomfunctie, de stroomlijnen niet volledig kan vangen, omdat het windveld  $\underline{V}$  uiteindelijk niet altijd divergentievrij is.

## 6. De methode van Picard

### 6.1. De oplossingstechniek (zie (1) blz. 9 en (4) blz. 194).

Hierbij wordt een stroomlijn door een gegeven beginpunt  $(x_0, y_0)$  berekend. Stellen we  $f(x,y) = v(x,y)/u(x,y)$  en integreren we (2.1) in de x-richting, dan krijgen we:

$$\int_{x_0}^{x_1} \frac{dy}{dx} dx = \int_{x_0}^{x_1} f(x,y) dx \quad (6.1.1)$$

Omdat we  $f(x,y)$  slechts kennen in de roosterpunten van een rechthoekig netwerk, zie punt 2, vervangen we (6.1.1) door

$$y_1 - y_0 = \frac{h}{2} \{f(x_0, y_0) + f(x_1, y_1)\} \quad (6.1.2)$$

Hierbij is  $h$  de afstand waarover geïntegreerd wordt. Kiezen we nu  $y_0$  als eerste benadering voor  $y_1$ , dan vinden we met (6.1.2) een tweede benadering voor  $y_1$ . Door de tweede benadering van  $y_1$  wederom in (6.1.2) in te vullen vinden we een derde benadering voor  $y_1$  enz.

We gaan net zolang door totdat het verschil in absolute waarde tussen de  $(n+1)$ -ste en de  $n$ -de benadering kleiner is dan bv.  $5 \times 10^{-6}$  (waarbij dit laatste getal klein is t.o.v.  $y_0$  en  $y_1$ ).

Er wordt nu bij benadering aan formule (6.1.2) voldaan en we hebben, uitgaande van het punt  $(x_0, y_0)$ , het punt  $(x_1, y_1)$  van de stroomlijn berekend.

Met een soortgelijk iteratieproces kunnen we het volgende punt  $(x_2, y_2)$  bepalen uit het punt  $(x_1, y_1)$ , hiervoor geldt de formule:

$$y_2 - y_1 = \frac{h}{2} \{f(x_1, y_1) + f(x_2, y_2)\} \quad (6.1.3)$$

Algemeen geldt dat, als we het punt  $(x_j, y_j)$  berekenen uit  $(x_{j-1}, y_{j-1})$ :

$$y_j - y_{j-1} = \frac{h}{2} \{f(x_{j-1}, y_{j-1}) + f(x_j, y_j)\} \quad (6.1.4)$$

De waarden  $f(x_{j-1}, y_{j-1})$  en  $f(x_j, y_j)$  vinden we door lineaire interpolatie uit de waarden van  $u$  en  $v$  in de roosterpunten.

Hebben we eenmaal een stroomlijn berekend, dan kunnen we door het kiezen van een nieuw beginpunt een volgende stroomlijn bepalen.

Bij voorgaande methode dienen we de  $x$ -waarden te kiezen, bv. equidistant op de  $x$ -as.

We kunnen dan alleen stroomlijnen bepalen, die lopen van links naar rechts of omgekeerd; dit houdt natuurlijk een beperking in, omdat niet elke stroomlijn hier aan voldoet.

Bovendien moeten we oppassen dat de stroomlijnen niet te steil verlopen, in dit geval wordt nl.  $u(x, y)$  klein of zelfs nul, zodat  $f(x, y)$  zeer groot wordt en het iteratieproces niet meer convergeert.

In punt 6.2 wordt voorgaande methode uitgebreid, zodat praktisch elke stroomlijn kan worden bepaald.

## 6.2. Uitbreiding van de methode

We zoeken de oplossing van de differentiaal vergelijking

$$\frac{dy}{dx} = \frac{v(x, y)}{u(x, y)} = f(x, y) \quad (6.2.1)$$

Door numerieke integratie van (6.2.1) leiden we af (zie 6.1.4) dat:

$$y_j = y_{j-1} + \frac{h}{2} \{f(x_{j-1}, y_{j-1}) + f(x_j, y_j)\} \quad (6.2.2)$$

met  $h = x_j - x_{j-1}$

Formule (6.2.2) passen we alleen toe indien  $|\frac{dy}{dx}| < 1$ , omdat indien  $\frac{dy}{dx} (=f(x, y))$  groot wordt, het iteratieproces niet meer convergent is.

Voor het geval  $|\frac{dy}{dx}| > 1$  of  $|\frac{dx}{dy}| < 1$  schrijven we (6.2.1) a.v.:

$$\frac{dx}{dy} = \frac{u(x, y)}{v(x, y)} = g(x, y) \quad (6.2.3)$$

waarbij  $g(x, y) = 1/f(x, y)$ .

Numerieke integratie van (6.2.3) geeft analoog aan (6.2.2) dat:

$$x_j = x_{j-1} + \frac{h}{2} \{g(x_{j-1}, y_{j-1}) + g(x_j, y_j)\} \quad (6.2.4)$$

waarbij  $h = y_j - y_{j-1}$ .

Het bepalen van de x-coördinaten gaat op dezelfde manier als het berekenen van y-coördinaten bij punt 6.1.

Bij het berekenen van een volgend punt van een stroomlijn uit een voorgaande punt  $(x,y)$  kunnen zich vier mogelijkheden voordoen nl.:

I :  $|dy/dx| \leq 1$  met  $u > 0$ .

In dit geval wordt een nieuwe y berekend op de lijn  $x + h$  (zie fig. 3)

II :  $|dy/dx| \leq 1$  met  $u < 0$ .

Een nieuwe y wordt berekend op de lijn  $x - h$ .

III :  $|dx/dy| < 1$  met  $v > 0$ .

Een nieuwe x wordt berekend op de  $y + h$  lijn.

IV :  $|dx/dy| < 1$  met  $v < 0$ .

Een nieuwe x wordt berekend op een  $y - h$  lijn (zie fig. 3).

In alle vier de gevallen hebben de grootheden  $dy/dx$ ,  $dx/dy$ ,  $u$  en  $v$  betrekking op het punt  $(x,y)$ .

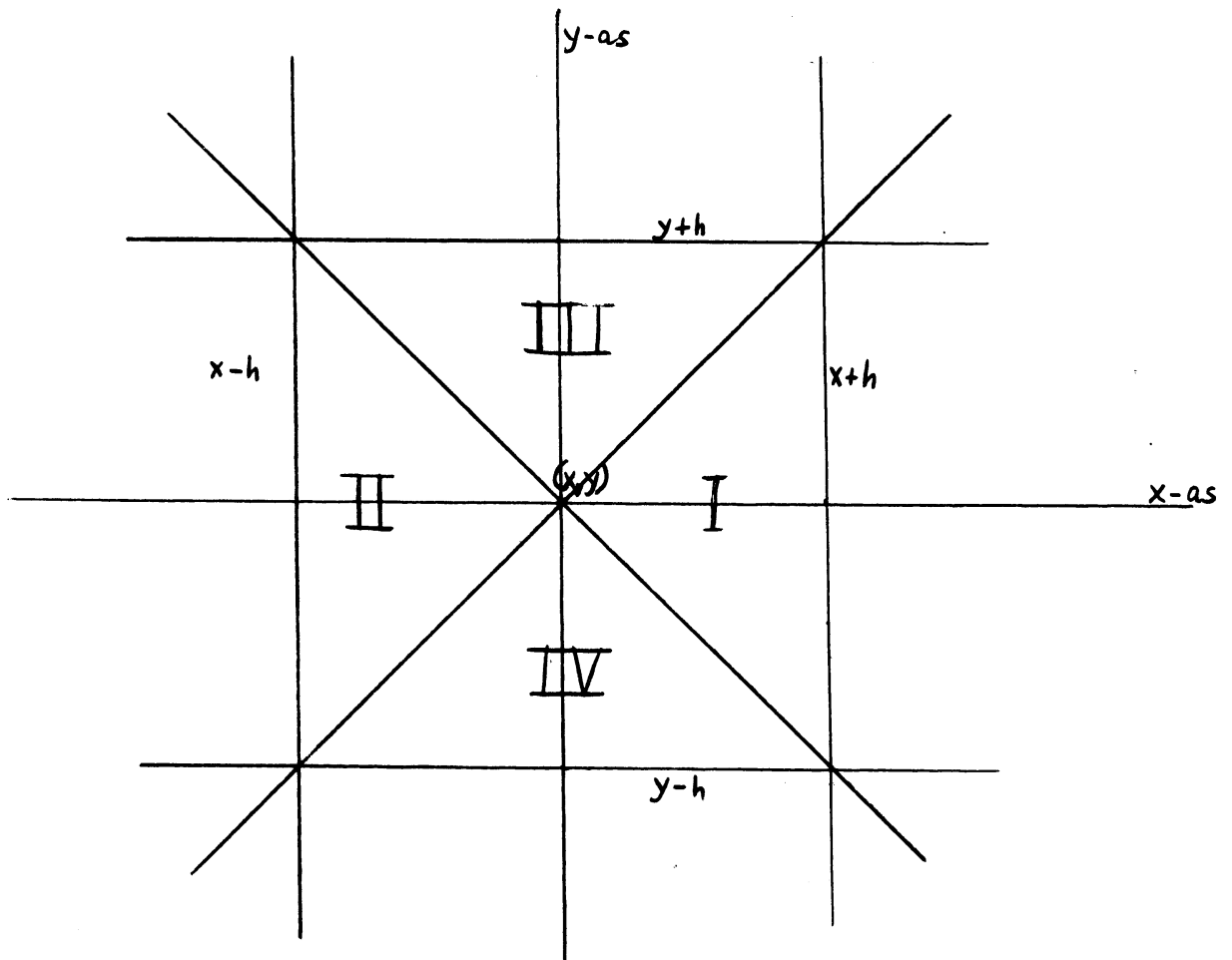
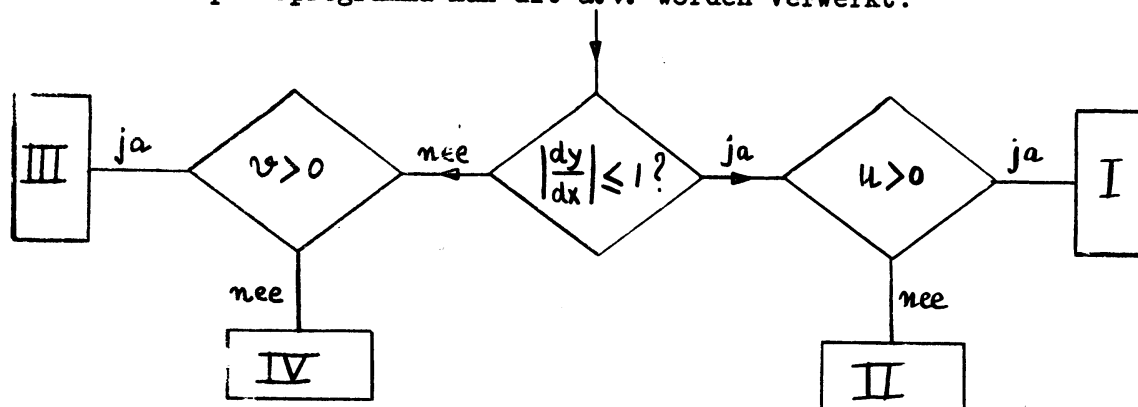


fig. 3

In het computerprogramma kan dit a.v. worden verwerkt:



In het programma wordt de  $h$  zodanig gekozen dat de  $x + h$  of  $x - h$  of  $y + h$  of de  $y - h$  lijn steeds de dichtstbijzijnde roosterlijn is.

Indien de verandering in de richting van de stroomlijn te groot wordt convergeert het iteratieproces (zie 6.1) niet meer, met als gevolg dat de stroomlijn niet verder berekend kan worden. Dit lichten we toe door een voorbeeld.

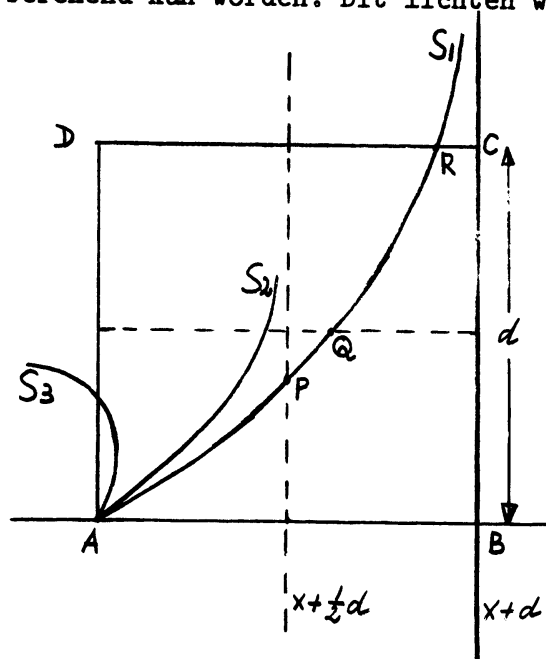


fig. 4

In de roosterpunten A, B, C en D van een rechthoekig rooster met roosterpuntsafstand  $d$  (zie fig. 4), zijn  $u$  en  $v$  gegeven.

Door het punt  $A = (x, y)$  loopt een stroomlijn  $S_1$ .

In het punt  $(x, y)$  is  $|dy/dx| < 1$  en  $u > 0$ . Willen we nu het volgende punt bepalen van de stroomlijn  $S_1$  op de lijn  $x + d$ , dan zien we dat dit niet mogelijk is, omdat de stroomlijn de lijn  $x + d$  niet snijdt.

Teneinde de stroomlijn S1 toch verder te kunnen berekenen, bepalen we het volgende punt op de lijn  $x + \frac{1}{2}d$  met het genoemde iteratieproces. Uit het punt P waar  $|dx/dy| < 1$  en  $v > 0$ , kunnen we vervolgens met formule (6.2.4) het punt Q bepalen en uit het punt Q, het punt R enz.

Om aan te geven op welke lijn we het volgende punt van de stroomlijn wensen te bepalen, voeren we de grootte  $h_0$  in. Voor  $h_0 = 1$  bepalen we het volgende punt op de  $x + d$  lijn, voor  $h_0 = 0.5$  op de  $x + \frac{1}{2}d$  lijn, voor  $h_0 = 0.1$  op de  $x + 0.1d$  lijn enz.

Voor  $h_0$  geldt echter steeds:  $0 < h_0 \leq 1$  (zie ook stroomdiagram).

Voor de stroomlijn S2 is de verandering in de richting van de stroomlijn nog groter dan bij S1. In dit geval snijdt S2 zelfs niet de lijn  $x + \frac{1}{2}d$ , zodat het iteratieproces niet zal convergeren. Om het rekenproces toch doorgang te doen vinden, wordt een iteratieteller meegegeven. Bereikt deze een bepaalde vastgestelde waarde (in de praktijk is 25 voldoende) dan breken we het iteratieproces af. Het vaststellen van het maximale aantal toegestane iteraties op 25, zal het normale verloop van het rekenproces niet beïnvloeden, omdat dit alleen in werking treedt als het iteratieproces niet of slechts langzaam convergeert.

Wensen we van de stroomlijn S2 toch het volgende punt te bepalen, dan kunnen we nog proberen het snijpunt te vinden met de  $y + \frac{1}{2}d$  lijn of de  $y - \frac{1}{2}d$  lijn. In dit geval kiezen we de  $y + \frac{1}{2}d$  lijn, omdat in het punt  $(x,y)$ ,  $v$  positief is. Het is natuurlijk ook mogelijk  $h_0 = \frac{1}{4}$  te stellen en dus het snijpunt te bepalen met de  $x + \frac{1}{4}d$  lijn.

Voor  $h_0 = \frac{1}{2}$  is het niet mogelijk van de stroomlijn S3 het volgende punt te bepalen, omdat deze stroomlijn de lijn  $x + \frac{1}{2}d$  niet snijdt, zelfs niet de lijn  $y + \frac{1}{2}d$ . Het iteratieproces zal dan niet convergeren en de stroomlijn wordt niet verder bepaald. We nemen dan een nieuw beginpunt om zodoende een andere stroomlijn te bespreken.

Voor het geval  $h_0 = \frac{1}{2}$  zal de zuivere rekentijd groter worden en wel ongeveer tweemaal groter per stroomlijn.

Voor  $h = \frac{1}{4}$  zal de zuivere rekentijd ongeveer viermaal groter worden per stroomlijn.

Voor een foutenbeschouwing verwijzen we naar 6.3.

### 6.3. Foutenbeschouwing

Mogelijke fouten die bij de Picard-methode kunnen optreden zijn:

#### 1. Inherente fouten

Dit zijn fouten die voortkomen uit foute gegevens, dus bv. waarnemings- en copierfouten.

Grote waarnemingsfouten en copieerfouten zijn er manueel uit te halen. Kleinere waarnemingsfouten blijven echter aanwezig.

2. Discretiseringsfouten

Deze fouten worden ook wel genoemd afbreek- of procesfouten.

Door de integraal in het rechterlid van formule (6.1.1) te vervangen door een numerieke benadering m.b.v. de trapeziumregel, wordt een discretiseringsfout gemaakt, die we voorlopig gelijk stellen aan  $E_d$ .

3. Interpolatiefouten

De fout die gemaakt wordt door lineair te interpoleren stellen we  $E_i$ .

4. Afrondingsfouten

Aangezien de EL-X8 in 12 decimalen nauwkeurig werkt, dus een maximale fout van  $0.5_{10}^{-12}$  voor één bewerking, zal de fout voor een groot aantal bewerkingen nog klein zijn.

5. Fouten, die gemaakt worden door bij het iteratieproces een tolerantie toe te staan

Wordt bij het iteratieproces (zie 6.1) geëist dat het verschil in absolute waarde tussen de (n+1)-ste en n-de benadering kleiner is dan  $0.5_{10}^{-5}$ , dan zal de fout na 100 stappen kleiner zijn dan  $0.5_{10}^{-3}$ .

D.w.z. dat het resultaat in drie decimalen nauwkeurig bepaald wordt. De bijdrage van dit soort fouten is in de praktijk te verwaarlozen.

Het is duidelijk dat de discretiserings- en interpolatiefouten samen met de waarnemingsfouten de belangrijkste rol spelen.

In de formule (6.1.4)

$$y_j - y_{j-1} = \frac{h}{2} \{f(x_{j-1}, y_{j-1}) + f(x_j, y_j)\} \quad (6.3.1)$$

wordt een discretiseringsfout gemaakt, die gelijk is aan  $E_d$ .

Doordat  $f(x_{j-1}, y_{j-1})$  en  $f(x_j, y_j)$  verkregen worden door lineaire interpolatie wordt in beide grootheden een interpolatiefout gemaakt. De totale interpolatiefout in het rechterlid van formule (6.3.1) is dus maximaal

$$\frac{h}{2} \cdot 2 |E_i| = h |E_i|$$

In  $f(x_{j-1}, y_{j-1})$  en  $f(x_j, y_j)$  zitten ook waarnemingsfouten, die we gelijk aan  $E_w$  stellen. De totale maximale waarnemingsfout in het rechterlid van (6.3.1) is dus

$$\frac{h}{2} \cdot 2 |E_w| = h |E_w|$$

De totale maximale fout is

$$E_j = |E_d| + h |E_i| + h |E_w| \quad (6.3.2)$$

Wordt dus een volgend punt van de stroomlijn bepaald, dan is de maximale fout  $E_j$ . Omdat telkens het volgende punt uit het voorgaande berekend wordt en de fouten zich dus accumuleren, is de maximale fout na  $n$  stappen

$$E_n = \sum_{j=1}^n E_j \quad (6.3.3)$$

Substitutie van (6.3.2) in (6.3.3) geeft:

$$E_n = \sum_{j=1}^n (|E_d| + h|E_i| + h|E_w|) \quad (6.3.4)$$

In de praktijk zal de fout natuurlijk aanmerkelijk kleiner zijn dan de max. fout  $E_n$ , omdat  $E_i$ ,  $E_d$  en  $E_w$  zowel positief als negatief kunnen zijn. De interpolatiefout  $|E_i|$  is maximaal  $|M_j|d^2$ , waarbij  $M_j$  afhankelijk is van  $\partial^2 u(x, \xi_j) / \partial y^2$  en  $\partial^2 v(x, \xi_j) / \partial y^2$  of van  $\partial^2 u(\xi_j, y) / \partial x^2$  en  $\partial^2 v(\xi_j, y) / \partial x^2$  en  $d$  de roosterpuntsafstand voorstelt;  $\xi_j$  ligt tussen de eindpunten van het interpolatie-interval.

De discretiseringsfout  $|E_d|$  is maximaal  $|N_j|h^3$ , waarbij  $N_j$  afhankelijk is van de tweede afgeleide van  $f(x, y)$  of  $g(x, y)$ .

De maximale fout na  $n$  stappen voor een stroomlijn wordt nu, indien we  $|E_i|$  en  $|E_d|$  substitueren in (6.3.4):

$$E_n = \sum_{j=1}^n (|N_j|h^3 + h|M_j|d^2 + h|E_w|)$$

Voor  $h = d$  geldt:

$$E_n = d^3 \sum_{j=1}^n (|N_j| + |M_j|) + d \sum_{j=1}^n |E_w| \quad (6.3.5)$$

Verfijnen we het rooster door de roosterpuntsafstand te verminderen tot de helft, dan wordt het aantal stappen per stroomlijn verdubbeld en is de maximale fout aan het einde van de stroomlijn

$$E_{2n} = \frac{1}{8} d^3 \sum_{j=1}^{2n} (|N_j| + |M_j|) + \frac{1}{2} d \sum_{j=1}^{2n} |E_w| \quad (6.3.6)$$

De bijdrage van de discretiserings- en interpolatiefout in de maximale fout is viermaal kleiner geworden, terwijl de bijdrage van de waarnemingsfout ongewijzigd is.

De rekentijd wordt per stroomlijn ongeveer tweemaal groter, omdat het aantal stappen om de stroomlijn te berekenen tweemaal zo groot wordt. Dat de rekentijd niet precies tweemaal zo groot wordt, komt doordat het aantal iteraties per stap kleiner is, omdat de beginwaarde beter aansluit bij de eindwaarde.



In het voorgaande is steeds  $h_0 = 1$  verondersteld, zodat de berekende punten van de stroomlijnen steeds op de roosterlijnen liggen.

Anders wordt het wanneer we b.v.  $h_0 = 0.5$  kiezen.

Dan wordt namelijk de interpolatiefout  $|E_i|$  groter. Stellen we het maximum van  $|E_i|$  gelijk aan  $2 |M_j| d^2$ , dus tweemaal de interpolatiefout van het oorspronkelijke geval ( $h_0 = 1$ ), dan zitten we beslist aan de veilige kant.

Dat de interpolatiefout groter wordt lichten we toe m.b.v. fig. 5.

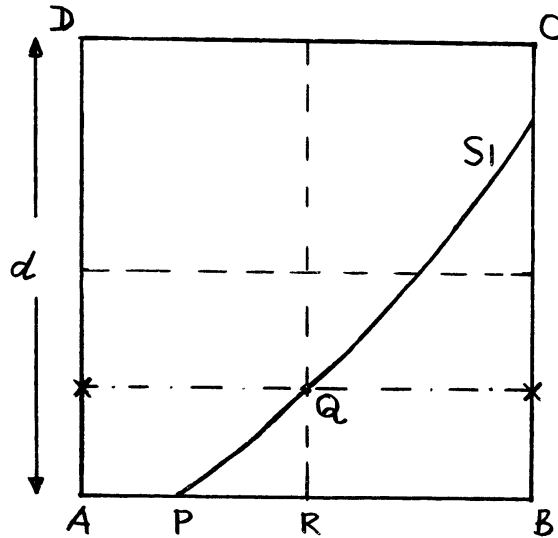


fig. 5

In de roosterpunten A, B, C en D zijn u en v gegeven. De stroomlijn S1 gaat door het punt P. Aangezien  $h_0 = 0.5$  bepalen we het volgende punt van de stroomlijn op de lijn QR. De waarde van  $f(x,y)$  in bv. het punt Q kan nu niet meer rechtstreeks verkregen worden door lineaire interpolatie van de waarden van u en v in de roosterpunten.

De waarde van  $f(x,y)$  in het punt Q wordt verkregen door lineair te interpoleren tussen de twee punten die door een kruisje zijn aangegeven (zie fig. 5). Maar deze punten zijn reeds gevonden door te interpoleren tussen twee roosterpunten, zodat dus de interpolatiefout in het punt Q groter zal zijn dan bv. in het punt R.

Voor  $h_0 = 0.5$  zal het aantal stappen per stroomlijn ongeveer tweemaal groter zijn dan voor  $h_0 = 1$ , zodat voor de maximale fout aan het einde van de stroomlijn volgens (6.3.4) geldt:

$$E_{2n} = \sum_{j=1}^{2n} (|N_j| h^3 + h \cdot 2 |M_j| d^2 + h |E_w|) \quad (6.3.7)$$

Voor  $h_0 = \frac{1}{2}$  is h echter maximaal  $\frac{1}{2}d$ , zodat voor de maximale fout geldt:

$$E_{2n} = \frac{1}{8} d^3 \sum_{j=1}^{2n} (|N_j| + 8 |M_j|) + \frac{1}{2} d \sum_{j=1}^{2n} |E_w| \quad (6.3.8)$$

Vergelijken we (6.3.8) met (6.3.5) dan zien we dat de bijdrage van de discretiseringsfout viermaal kleiner is geworden, terwijl de bijdrage van de interpolatiefout ongeveer tweemaal groter is dan in het oorspronkelijke geval (zie 6.3.5).

De bijdrage van de waarnemingsfout blijft ongewijzigd.

Indien de bijdrage van de discretiseringsfout en de interpolatiefout van dezelfde orde van grootte zijn (zie 6.3.5), mogen we wel stellen dat de totale maximale fout niet veel afwijkt van de totale maximale fout in het oorspronkelijke geval.

Voor  $h_0 = 0.1$  wordt de maximale fout aan het einde van de stroomlijn:

$$E_{10n} = d^3 \sum_{j=1}^{10n} \left( \frac{1}{1000} |N_j| + \frac{1}{5} |M_j| \right) + \frac{d}{10} \sum_{j=1}^{10n} |E_w| \quad (6.3.9)$$

Ook hier wijkt de totale maximale fout niet veel af van die in (6.3.5).

#### 6.4. Toelichting bij het stroomdiagram

Wensen we de Picard-methode geschikt te maken voor de verwerking met een computer, door er een programma voor te schrijven, dan is het raadzaam eerst een stroomdiagram op te stellen, dat de structuur van het programma weergeeft. Om het verloop van het stroomdiagram beter te kunnen volgen, wordt eerst een toelichting bij het stroomdiagram gegeven. In de roosterpunten van een rechthoekig netwerk (p,q) zijn u en v gegeven, zie fig. 6.

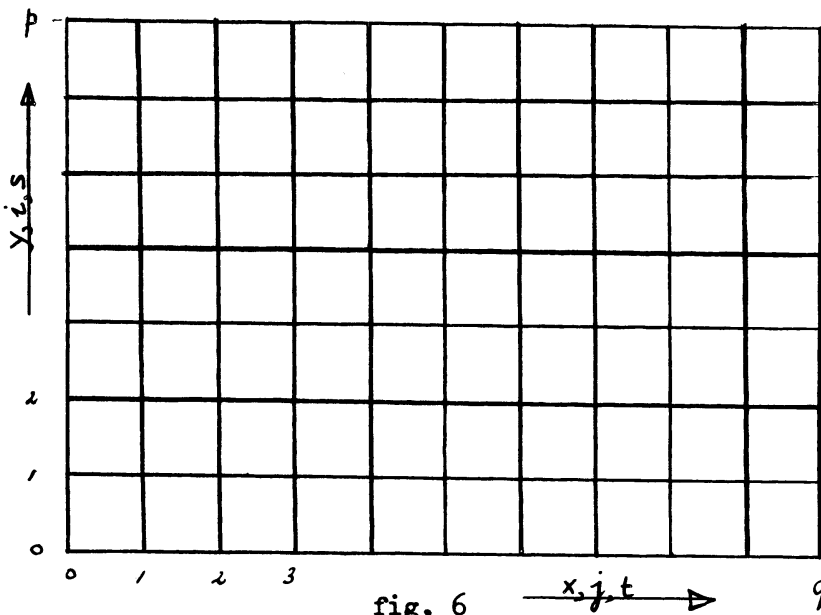


fig. 6

De roosterpuntsafstand stellen we gelijk aan één ( $d=1$ ). In werkelijkheid heeft de roosterpuntsafstand natuurlijk een andere waarde, maar het stellen van  $d=1$  heeft geen gevolgen voor het verloop van de stroomlijnen in het netwerk. Dit houdt nog niet in dat de berekeningen onafhankelijk van de roosterpuntsafstand zijn. Het netwerk wordt alleen met een factor  $d$  verkleind. Na vermenigvuldiging met de factor  $d$  kunnen we de werkelijke plaats van de stroomlijnen vastleggen, zodra deze zijn berekend. Het stellen van  $d=1$  komt neer op schaling van de  $x$  en  $y$  waarden. Het voordeel van het stellen van  $d=1$  is, dat we in het programma niet steeds met  $d$  behoeven te vermenigvuldigen, resp. door  $d$  behoeven te delen.

De connector VS duidt aan dat we een volgende stroomlijn gaan berekenen, terwijl de connector VP wil zeggen dat we het volgende punt van de stroomlijn gaan bepalen.

Het is duidelijk dat, zodra de stroomlijn buiten het netwerk komt, deze niet verder berekend kan worden en de berekening van de desbetreffende stroomlijn dient te worden afgebroken. Dit vergt echter extra voorzorgsmaatregelen voor het programma.

Dat het iteratieproces niet convergeert kan blijken doordat de iteratieteller 25 wordt of doordat de berekende  $x$  of  $y$  buiten de roostergrenzen treedt.

Voor het stroomdiagram verwijzen we naar bijlage 1.

Voor het geval dat de stroomlijn een gesloten kromme is, binnen het rechthoekig netwerk, wordt de stroomlijn niet berekend, indien we als beginpunt een punt op de rand nemen. We dienen dan een beginpunt op de gesloten kromme te nemen. De moeilijkheid is echter dat we van te voren de ligging van de stroomlijnen niet kennen. We kiezen daarom net zo lang beginpunten - ook beginpunten die niet op de rand liggen - totdat het stroomlijnenbeeld voldoende bekend is.

Bij het berekenen van stroomlijnen die een gesloten kromme vormen doet zich nog het probleem voor, nl. uitgaande van het gekozen beginpunt komen we na een aantal stappen weer in dat punt terug.

Nemen we geen maatregelen dan zal het computerprogramma steeds verder rekenen. Na iedere stap testen we daarom of we misschien in het beginpunt terug gekomen zijn. Omdat er bij het berekenen van de stroomlijnen fouten gemaakt worden (zie 6.3) zal een gesloten stroomlijn natuurlijk nooit exact in het beginpunt terugkeren. Daarom leggen we een gebiedje om het beginpunt (zie het stroomdiagram) en testen we, nadat een aantal stappen zijn berekend, of de stroomlijn wellicht weer binnen genoemd gebiedje komt. Is dit het geval dan wordt de stroomlijn niet verder berekend.

Teneinde aan te geven in welke richting we de stroomlijnen willen berekenen, voeren we een grootheid  $R$  in, die  $+1$  of  $-1$  kan zijn. Indien  $R = +1$  dan wordt het volgende punt van de stroomlijn bepaald in de richting van de vector  $(u, v)$  en als  $R = -1$  berekenen we het volgende punt in de richting van de vector  $(-u, -v)$ .

De reden dat we deze grootheid  $R$  invoeren zal duidelijk worden als we kijken naar fig. 12. We kunnen hier de stroomlijnen het beste bepalen in de richting van de werveling, waardoor we  $R = -1$  moesten stellen.

In principe kunnen we de beginpunten willekeurig kiezen en door elk gekozen beginpunt een stroomlijn berekenen.

De vraag is echter, welke beginpunten kiezen we, omdat we het stroomlijnen-beeld van het gegeven veld zo snel en zo goed mogelijk kunnen bepalen. Het is nog niet mogelijk hier een objectieve methode voor aan te geven. Een globale indruk over het verloop van de stroomlijnen is daarom zeer nuttig en zelfs gewenst, om de beginpunten zo goed mogelijk te kiezen. Blijkt echter dat het geplote veld nog niet voldoende bepaald is, dan kunnen we, alsnog nieuwe beginpunten invoeren.

Met de grootheid  $h_0$ , waarvoor geldt  $0 < h_0 \leq 1$ , kunnen we aangeven op welke horizontale of verticale lijn we het volgende punt van de stroomlijn wensen te bepalen.

Voor het geval  $h_0 = 1$  zal het volgende berekende punt van de stroomlijn steeds op een roosterlijn liggen.

#### 6.5. Het programma SPM

Het programma is geschreven in ALGOL voor de EL-X8 rekenmachine.

De invoergegevens, die op ponsband staan, bestaan uit:

1. de waarden van  $p$  en  $q$ , die het netwerk vastleggen;
2. de waarde van  $R$ , die  $+1$  of  $-1$  is;
3. de waarde  $h_0$ , waarbij  $0 < h_0 \leq 1$ ;
4. het nummer van het vectorveld;
5. de waarden van  $u$  en  $v$  is de roosterpunten;
6. de coördinaten van het beginpunt van elke te berekenen stroomlijn en
7. het getal  $-1$ , ter aanduiding dat de onder 6 vermelde serie beginpunten is geëindigd en waarmee de berekening wordt beëindigd.

De uitvoergegevens bestaan uit de begin- en eindpunten van elke berekende stroomlijn. De stroomlijnen worden met de plotter getekend. Voor het algol-programma zij verwezen naar bijlage 2.

6.6. Toepassingen van het programma SPM, waarvan de analytische oplossing bekend is  
Teneinde een indicatie te verkrijgen omtrent de fout die gemaakt wordt met de uitgebreide PICARD-methode zijn eerst enkele voorbeelden met genoemde methode verwerkt, waarvan de analytische oplossing bekend is.

Voorbeeld 1:

De differentiaal vergelijking  $dy/dx = v(x,y)/u(x,y)$  heeft de vorm:

$$\frac{dy}{dx} = \frac{-x}{y}$$

met  $v(x,y) = -x$  en  $u(x,y) = y$  (6.6.1)

De oplossing van deze diff. vgl. is:  $x^2 + y^2 = r^2$

De stroomlijnen zijn dus cirkels, concentrisch om de oorsprong.

De componenten  $u$  en  $v$  van de vectoren in de roosterpunten zijn m.b.v. (6.6.) gemakkelijk te berekenen.

Voorbeeld 2:

$$\frac{dy}{dx} = 1 \text{ met } v(x,y) = 1 \text{ en } u(x,y) = 1.$$

De oplossing van de diff.vgl. is:  $y = x + C$

d.w.z. de stroomlijnen zijn rechte lijnen.

Voorbeeld 3:

$$\frac{dy}{dx} = 2x \text{ met } v(x,y) = -2x \text{ en } u(x,y) = -1.$$

De oplossing is:  $y = x^2 + C$ .

Voorbeeld 4:

$$\frac{dy}{dx} = 3x^2 \text{ met } v(x,y) = -3x^2/100 \text{ en } u(x,y) = -1/100.$$

De oplossing is  $y = x^3 + C$ .

Voorbeeld 5:

$$\frac{dy}{dx} = \sin x \text{ met } v(x,y) = -\sin x \text{ en } u(x,y) = -1.$$

De oplossing van de diff.vgl. is:  $y = -\cos x + C$ .

Voorbeeld 6:

$$\frac{dy}{dx} = e^x \text{ met } v(x,y) = -e^x \text{ en } u(x,y) = -1.$$

De oplossing is  $y = e^x + C$ .

In de tabel hierna staan de fouten vermeld, die in de voorbeelden 1 t/m 6 optreden. Voor elk veld wordt slechts één stroomlijn berekend met beginpunt zoals in de tabel staat aangegeven. Voor ieder voorbeeld geldt  $p = q = 10$  en  $R = -1$ .

TABEL 1

| Voorbeeld | Beginpunt |    | Eindpunt van de stroomlijn |        |                                  |         |                           |         | Fout aan het einde van de stroomlijn |         |                   |         |
|-----------|-----------|----|----------------------------|--------|----------------------------------|---------|---------------------------|---------|--------------------------------------|---------|-------------------|---------|
|           |           |    | Analytisch                 |        | Numeriek met $h_0 = \frac{1}{2}$ |         | Numeriek met $h_0 = 1/10$ |         | Voor $h_0 = \frac{1}{2}$             |         | Voor $h_0 = 1/10$ |         |
|           |           |    |                            |        |                                  |         |                           |         | in x                                 | in y    | in x              | in y    |
| 1         | 10        | 0  | 0                          | 10     | 0.0000                           | 9.9983  | 0.0000                    | 10.0000 | 0                                    | 0.0017  | 0                 | 0       |
| 2         | 10        | 10 | 0                          | 0      | 0.0000                           | 0.0000  | 0.0000                    | 0.0000  | 0                                    | 0       | 0                 | 0       |
| 3         | 0         | 0  | 3.1623                     | 10     | 3.1645                           | 10.0000 | 3.1625                    | 10.0000 | -0.0022                              | 0       | -0.0002           | 0       |
| 4         | 0         | 0  | 2.1544                     | 10     | 2.0830                           | 10.0000 | 2.0796                    | 10.0000 | +0.0714                              | 0       | +0.0748           | 0       |
| 5         | 0         | 5  | 10                         | 6.5423 | 10.0000                          | 6.6833  | 10.0000                   | 6.6833  | 0                                    | -0.1410 | 0                 | -0.1410 |
| 6         | 0         | 1  | 2.3026                     | 10     | 2.2363                           | 10.0000 | 2.2352                    | 10.0000 | +0.0663                              | 0       | +0.0674           | 0       |

Onder fout verstaan we de werkelijke waarde verminderd met de benaderde waarde, of te wel:

fout = analytische waarde - numerieke waarde.

De gevallen waar in de tabel voor de fout nul staat, wil zeggen dat  $|fout| < 0.00005$ .

Voorbeeld 7:

$dy/dx = -(x-7)/(y-7)$  met  $v(x,y) = -(x-7)$  en  $u(x,y) = y-7$

De oplossing van deze differentiaalvergelijking is:

$$(x-7)^2 + (y-7)^2 = r^2$$

De stroomlijnen zijn dus cirkels concentrisch om het punt (7,7).

Hier worden hele cirkels (gesloten stroomlijnen) berekend, in tegenstelling tot voorbeeld 1, waar slechts kwart cirkels worden berekend.

Tabel 2 geeft een overzicht van de opgetreden fouten voor verschillende waarden van  $h_0$ .

TABEL 2

| $h_0$ | Beginpunt |   | Eindpunt   |        |          |        | Fout   |      |
|-------|-----------|---|------------|--------|----------|--------|--------|------|
|       |           |   | Analytisch |        | Numeriek |        | in x   | in y |
| 1     | 14        | 7 | 14.0000    | 7.0000 | 13.9838  | 7.0000 | 0.0162 | 0    |
| 0.5   | 14        | 7 | 14.0000    | 7.0000 | 13.9967  | 7.0000 | 0.0033 | 0    |
| 0.2   | 14        | 7 | 13.9886    | 6.6000 | 13.9882  | 6.6000 | 0.0004 | 0    |
| 0.1   | 14        | 7 | 13.9886    | 6.6000 | 13.9885  | 6.6000 | 0.0001 | 0    |

Uit tabel 2 blijkt dat de fout kleiner wordt naarmate  $h_0$  kleiner wordt. Dat dit echter niet altijd het geval is kan men zien in tabel 1 (zie voorbeeld 4, 5 en 6).

Uit theoretische beschouwingen (zie 6.3) blijkt dat de orde van grootte van de maximale fout niet verandert indien  $h_0$  gewijzigd wordt: daarbij wordt echter verondersteld dat de bijdrage van de discretiseringsfout en de interpolatiefout van dezelfde orde van grootte zijn. Deze veronderstelling geldt echter niet bij de voorbeelden 1, 3 en 7 waar de interpolatiefouten en de waarnemingsfouten gelijk nul zijn. Dit houdt in dat de maximale fout kleiner wordt naarmate  $h_0$  kleiner wordt, omdat nu slechts de discretiseringsfouten een bijdrage leveren. Voor  $h_0 = 0.1$  zal de maximale fout ongeveer 100 x kleiner worden (6.3.9), zo ook de werkelijk opgetreden fout.

Bij voorbeeld 2 zijn bovendien de discretiseringsfouten gelijk nul, zodat de maximale fout gelijk nul is.

In fig. 15 zijn met de plotter stroomlijnen (cirkels) getekend, waarbij  $h_0 = 0.2$ .

#### 6.7. Toepassingen in de praktijk

Bij voorbeelden uit de praktijk zal i.h.a. de analytische oplossing niet bekend zijn. Het is dan ook moeilijk te zeggen hoe groot de gemaakte fouten zijn. Enig inzicht in de grootte van de fouten hebben we wel (zie 6.3 en 6.6) De Picard-methode wordt toegepast op de gemiddelde wind over het eerste kwartaal op de Noord-Atlantische Oceaan.

Aangenomen wordt dat de componenten  $u$  en  $v$  van de windvectoren in de roosterpunten van een rechthoekig rooster bekend zijn.

De roosterpuntsafstand is gelijk aan twee graden, gemeten langs de evenaar. We geven hier vier toepassingen, die met het programma SPM zijn verwerkt. Bij het eerste voorbeeld is de verandering in de richting van de stroomlijnen klein, terwijl bij het tweede, derde en vierde voorbeeld de verandering in de richting van de stroomlijnen in bepaalde gebieden groot is. Bij het derde voorbeeld treedt bovendien een werveling op. In het vierde voorbeeld, waarin twee wervelingen voorkomen, is het rooster ongeveer tweemaal groter dan dat van de voorgaande drie voorbeelden.

In de figuren 7, 9, 11 en 13 zijn de stroomlijnen langs subjectieve weg verkregen. In de figuren 8, 10, 12 en 14 worden de stroomlijnen berekend met een objectieve methode nl. met de methode van Picard. Elke stroomlijn is uit de berekende punten met de plotter getekend. Voor elke toepassing geldt dat  $h_0 = 0.2$  en  $R = -1$ .

Doordat de stroomlijnen onafhankelijk van elkaar berekend worden, is het gevaar aanwezig dat t.g.v. fouten in het rekenproces, de stroomlijnen elkaar zullen snijden. Bij de voorbeelden die hier verwerkt zijn met het programma SPM is dit nog niet voorgekomen, met uitzondering echter van die gebieden waar een vortex optreedt.

De vier genoemde voorbeelden zijn ook verwerkt voor  $h_0 = 0.1, 0.5$  en  $1.0$ . De resultaten van het eerste voorbeeld waren voor verschillende waarden van  $h_0$  nauwelijks afwijkend van elkaar.

Voor  $h_0 = 0.1$  werden ongeveer dezelfde resultaten verkregen als voor  $h_0 = 0.2$ . Indien we echter  $h_0 = 0.2$  kiezen zal het rekenproces sneller voltooid zijn, omdat het aantal stappen per stroomlijn tweemaal kleiner is, met als gevolg dat de rekentijd ongeveer tweemaal kleiner wordt.

Voor  $h_0 = 1$  waren de resultaten bij de voorbeelden 2, 3 en 4 beslist van mindere kwaliteit. Dit komt doordat voor  $h_0 = 1$  het iteratieproces (zie 6.1) niet convergeert - of niet naar de juiste oplossing convergeert -, indien de verandering in de richting van de stroomlijnen te groot wordt.

#### 6.8. Conclusie

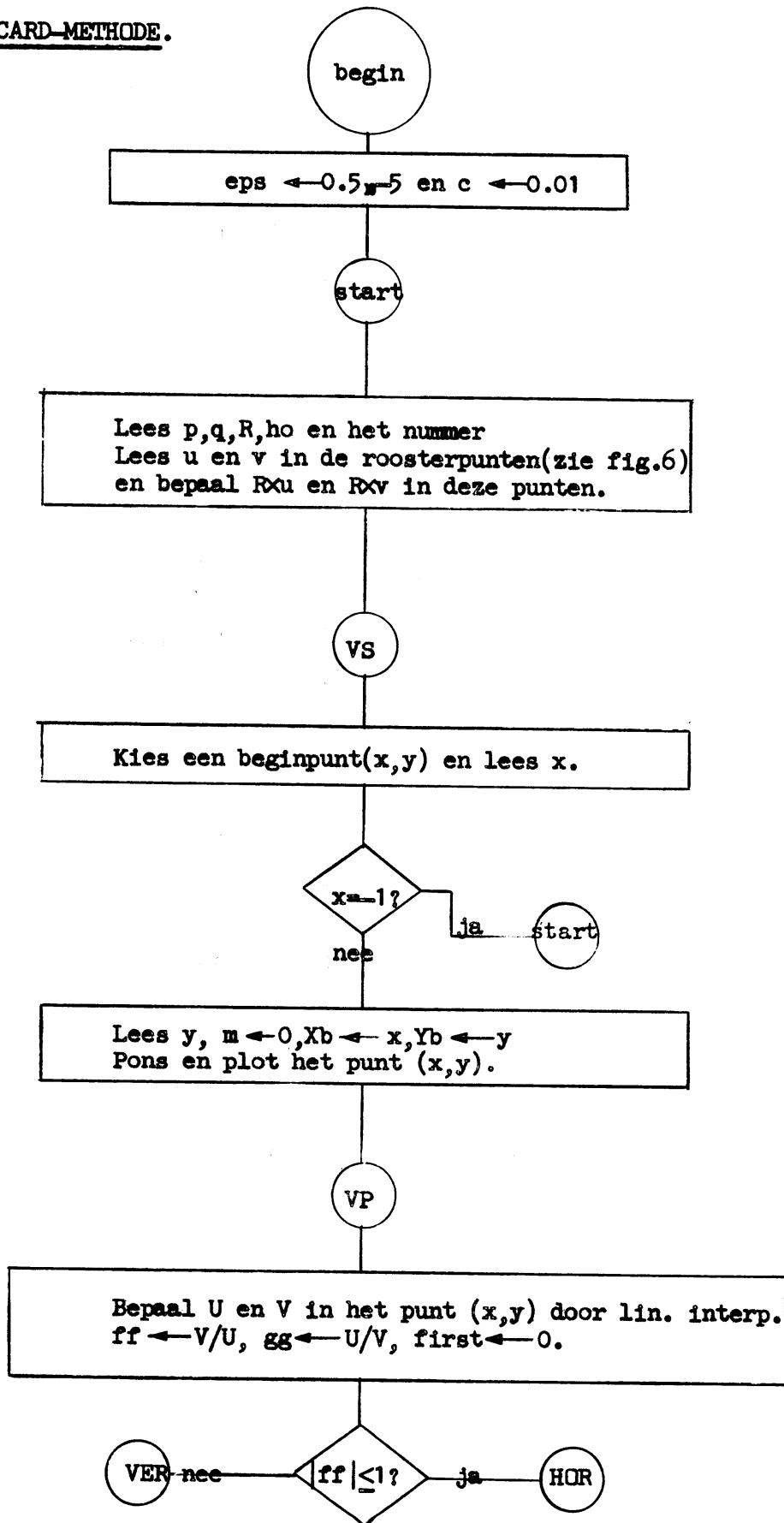
Met de uitgebreide methode van Picard is het mogelijk met de computer het verloop van de stroomlijnen te bepalen, indien de x en y componenten van een vectorveld gegeven zijn. Voor het computerprogramma SPM is het noodzakelijk dat de componenten bekend zijn in de roosterpunten van een rechthoekig rooster. Deze roosterwaarden zijn in de praktijk doorgaans niet bekend, zodat deze berekend dienen te worden uit de componenten in de waarnemingspunten.

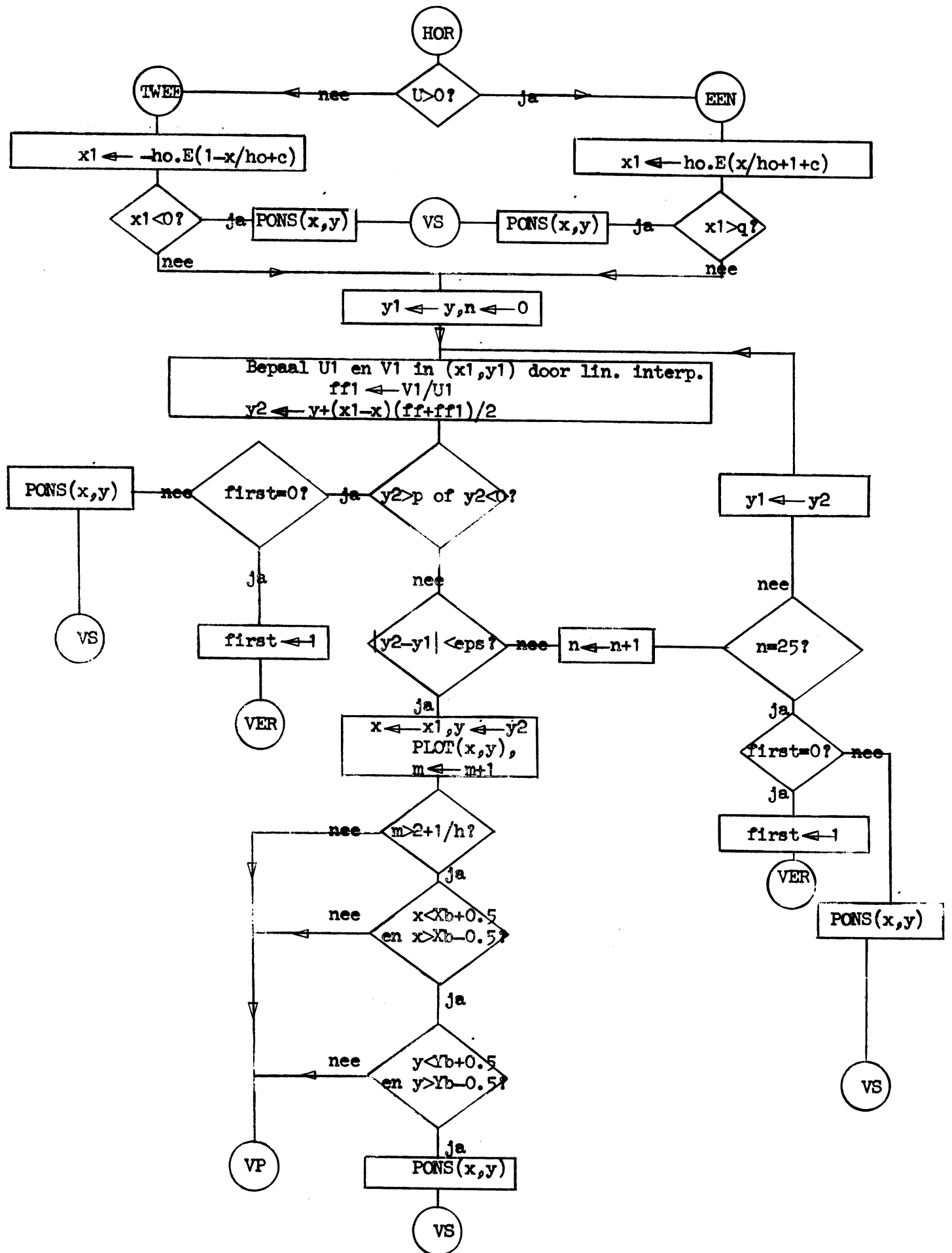
#### 7. Literatuur

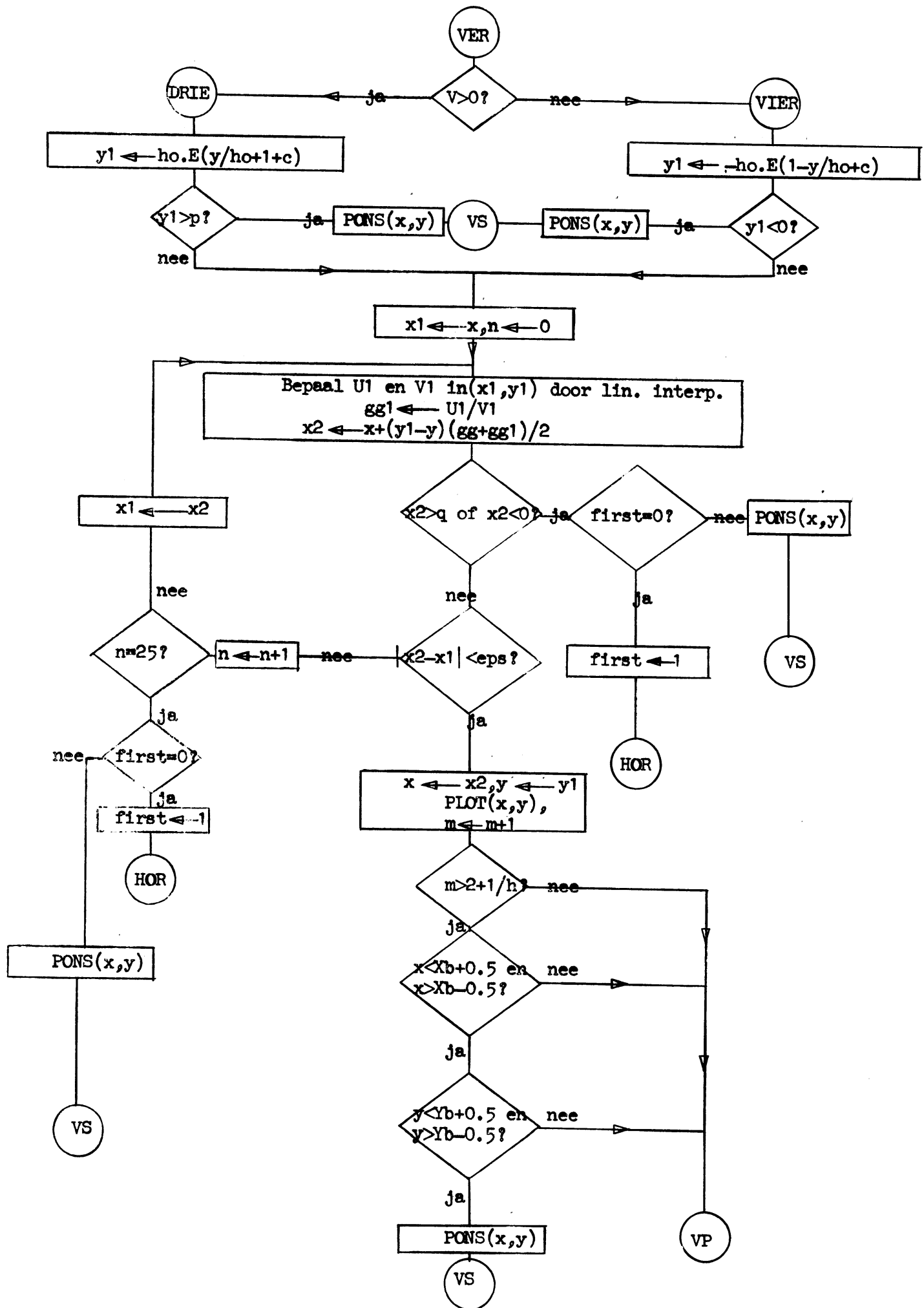
- (1) B. Heyna: Een objectieve methode voor het tekenen van stroomlijnkaarten, K.N.M.I., W.R. 62-3, 1962.
- (2) D.W. Pritchard: Streamlines from a discrete vector field; with application to ocean currents, Journal of Marine Research VII, 296-303, 1948.
- (3) H.F. Hawkins en S.L. Rosenthal: On the computation of stream functions from the wind field, M.W.R., vol. 93, no. 4, pp. 245-252, April 1965.
- (4) F.B. Hildebrand: Introduction to numerical analysis, 1956.



BIJLAGE I.  
STROOMDIAGRAM PICARD-METHODE.







BYLAGE 2

```

begin comment 221168-KNMI-CROT-SPM.
    Het berekenen van stroomlijnen door gegeven beginpunten
    volgens de PICARD-methode;
    integer p,q,k,i,j,m,n,s,t,R,xlast,ylast;
    real x,y,ff,gg,ff1,gg1,x2,y2,eps,Uxy,Vxy,Xb,Yb,h,x1,y1,Uxy1,Vxy1 ,c;
    boolean FIRST;
    eps:=0.5-5;
    c:=0.01;
START: RUNOUT; MOVE(16); p:=read; q:=read; R:=read; h:=read;
PUNLCR; PUNLCR; PUNLCR; PUTEXT(STROOMLYNEN NR. †); ABSFLXP(2,0,read);
PUNLCR;
begin real array u,v[0:p,0:q];

    procedure UITVOER;
    begin FLXP(4,4,x); FLXP(2,4,y); PUNLCR end;

    real procedure INTERP(Z,x,y); array Z; real x,y;
    begin real Z1,Z2;
        t:=entier(x); s:=entier(y);
        if t=q ^ s=p then INTERP:=Z[s,t] else
        if t=q then INTERP:=Z[s,t]+(y-s)×(Z[s+1,t]-Z[s,t]) else
        if s=p then INTERP:=Z[s,t]+(x-t)×(Z[s,t+1]-Z[s,t]) else
        begin Z1:=Z[s,t]+(y-s)×(Z[s+1,t]-Z[s,t]);
            Z2:=Z[s,t+1]+(y-s)×(Z[s+1,t+1]-Z[s,t+1]);
            INTERP:=Z1+(x-t)×(Z2-Z1)
        end
    end INTERP;

    procedure PLOT(X,Y); value X,Y; real X,Y;
    begin integer x,y,xy,dx,dy,dx2,dy2,i,dev;
        boolean right;
        x:=X/0.0254; y:=Y/0.0254;
        dx:=x-xlast; dy:=y-ylast; xlast:=x; ylast:=y; right:=dx>0;
        if dx<0 then begin dx:=-dx; x:=2 end else x:=1;
        if dy<0 then begin dy:=-dy; y:=8 end else y:=4; xy:=x+y;
        if dy>dx then begin x:=y; i:=dx; dx:=dy; dy:=i end;
        dx2:=dx×2; dy2:=dy×2; dev:=0;
        for i:=1 step 1 until dx do
        begin dev:=dev+dy2;
            if (if right then dev<dx else dev<dx) then MOVE(x) else
            begin MOVE(xy); dev:=dev-dx2 end
        end
    end PLOT;

    xlast := ylast := 0;
    for i:=1 step 1 until 600 do MOVE(1); MOVE(40);
    for i:=1 step 1 until 750 do MOVE(4); MOVE(32);
    for i:=q,0 do
    for j:=if i=0 then p else 0, if i=0 then 0 else p do PLOT(i,j);
    for i:=p step -1 until 0 do
    for j:=0 step 1 until q do
    begin u[i,j]:=R×read; v[i,j]:=R×read end;
VS: x:=read;
    if x=-1 then
    begin ABSFLXP(5,0,time); PUNLCR; MOVE(16); PLOT(q,0); goto START end;
    y:=read; m:=0; Xb:=x; Yb:=y;
    FLXP(4,4,x); FLXP(2,4,y); MOVE(16); PLOT(x,y); MOVE(32);
VP: Uxy:=INTERP(u,x,y); Vxy:=INTERP(v,x,y);
    ff:=Vxy/Uxy; gg:=Uxy/Vxy;
    FIRST:=true; if abs(ff)<1 then goto HOR else goto VER;

```

```

HOR:   if Uxy>0 then goto EEN else goto TWEE;
EEN:   x1:=hxentier(x/h+1+c);
       if x1>q then begin UITVOER; goto VS end else goto AAA;
TWEE:  x1:=hxentier(1-x/h+c);
       if x1<0 then begin UITVOER; goto VS end else goto AAA;
AAA:   y1:=y; n:=0;
ZPA:   Uxy1:=INTERP(u,x1,y1); Vxy1:=INTERP(v,x1,y1); ff1:=Vxy1/Uxy1;
       y2:=y+(x1-x)/2x(ff+ff1);
       if y2>p  $\vee$  y2<0 then
       begin   if FIRST then begin FIRST:=false; goto VER end else
           begin UITVOER; goto VS end
       end;
       if abs(y2-y1)<eps then goto VOLG else
       begin   n:=n+1; if n=25 then
           begin if FIRST then begin FIRST:=false; goto VER end else
               begin UITVOER; goto VS end
           end;
           y1:=y2; goto ZPA
       end;
VOLG:  end;
       x:=x1; y:=y2; PLOT(x,y);
       m:=m+1;
       if m>2+1/h then
       begin   if x<Xb+0.5  $\wedge$  x>Xb-0.5 then
           begin   if y<Yb+0.5  $\wedge$  y>Yb-0.5 then
               begin   UITVOER; PLOT(Xb,Yb); goto VS end
           end
       end
       end;
       goto VP;
VER:   if Vxy>0 then goto DRIE else goto VIER;
DRIE:  y1:=hxentier(y/h+1+c);
       if y1>p then begin UITVOER; goto VS end else goto BBB;
VIER:  y1:=hxentier(1-y/h+c);
       if y1<0 then begin UITVOER; goto VS end else goto BBB;
BBB:   x1:=x; n:=0;
ZPB:   Uxy1:=INTERP(u,x1,y1); Vxy1:=INTERP(v,x1,y1); gg1:=Uxy1/Vxy1;
       x2:=x+(y1-y)/2x(gg+gg1);
       if x2>q  $\vee$  x2<0 then
       begin   if FIRST then begin FIRST:=false; goto HOR end else
           begin UITVOER; goto VS end
       end;
       if abs(x2-x1)<eps then goto DOOR else
       begin   n:=n+1; if n=25 then
           begin   if FIRST then begin FIRST:=false; goto HOR end else
               begin UITVOER; goto VS end
           end;
           end; x1:=x2; goto ZPB
       end;
DOOR:  end;
       x:=x2; y:=y1; PLOT(x,y);
       m:=m+1;
       if m>2+1/h then
       begin   if x<Xb+0.5  $\wedge$  x>Xb-0.5 then
           begin   if y<Yb+0.5  $\wedge$  y>Yb-0.5 then
               begin   UITVOER; PLOT(Xb,Yb); goto VS end
           end
       end
       end; goto VP

```

end

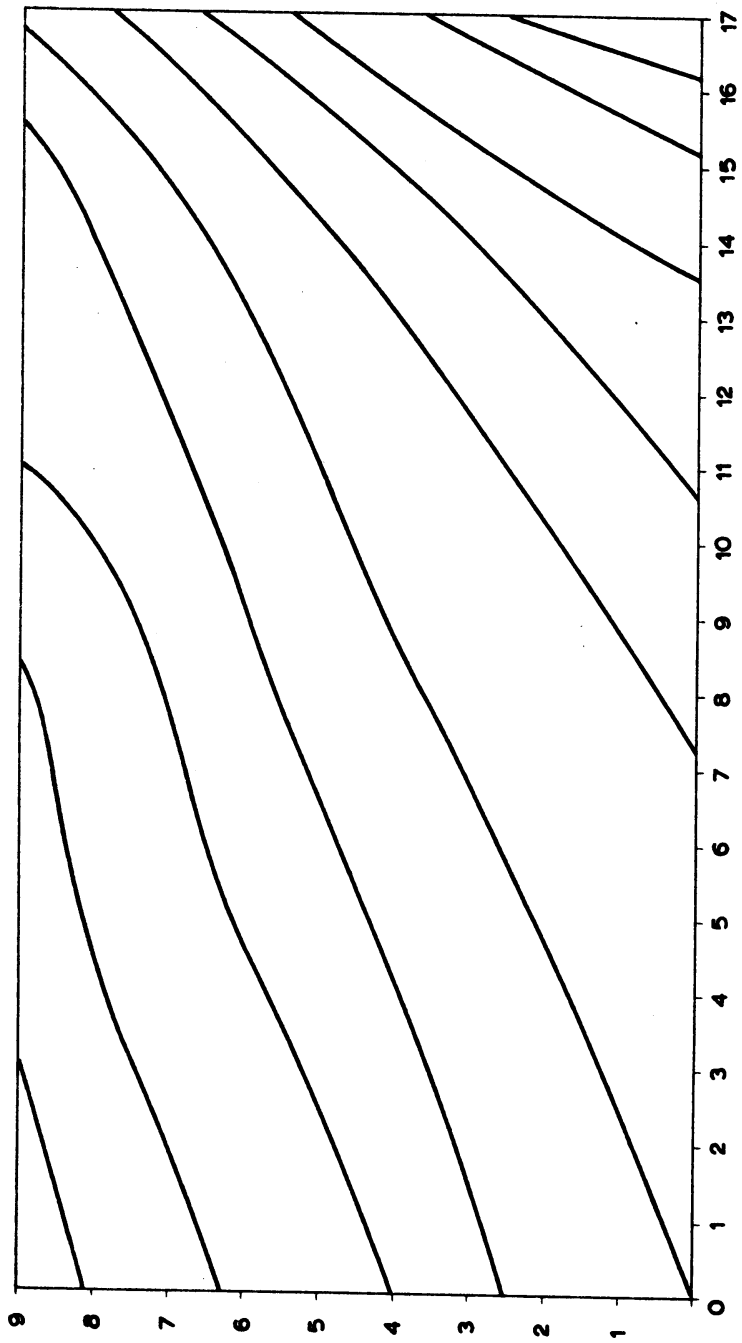
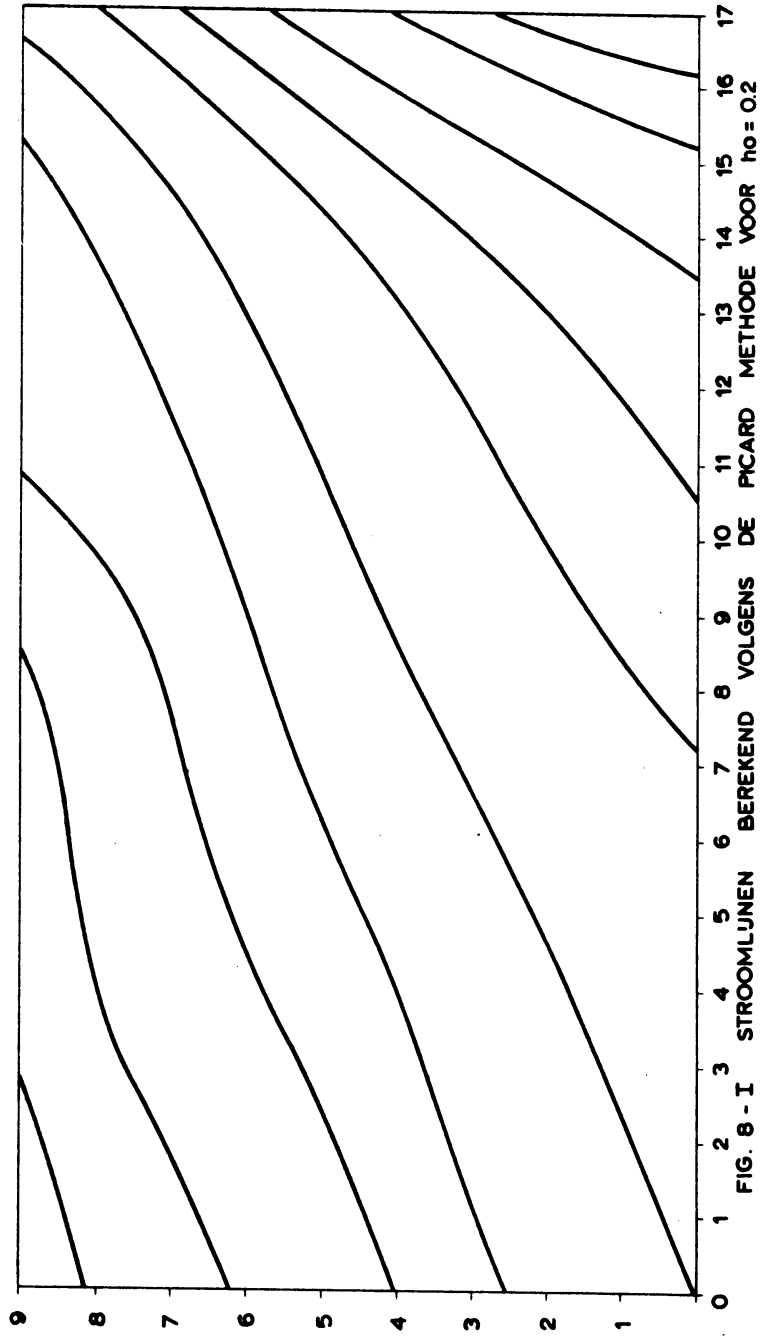


FIG. 7 - I STROOMLIJNEN



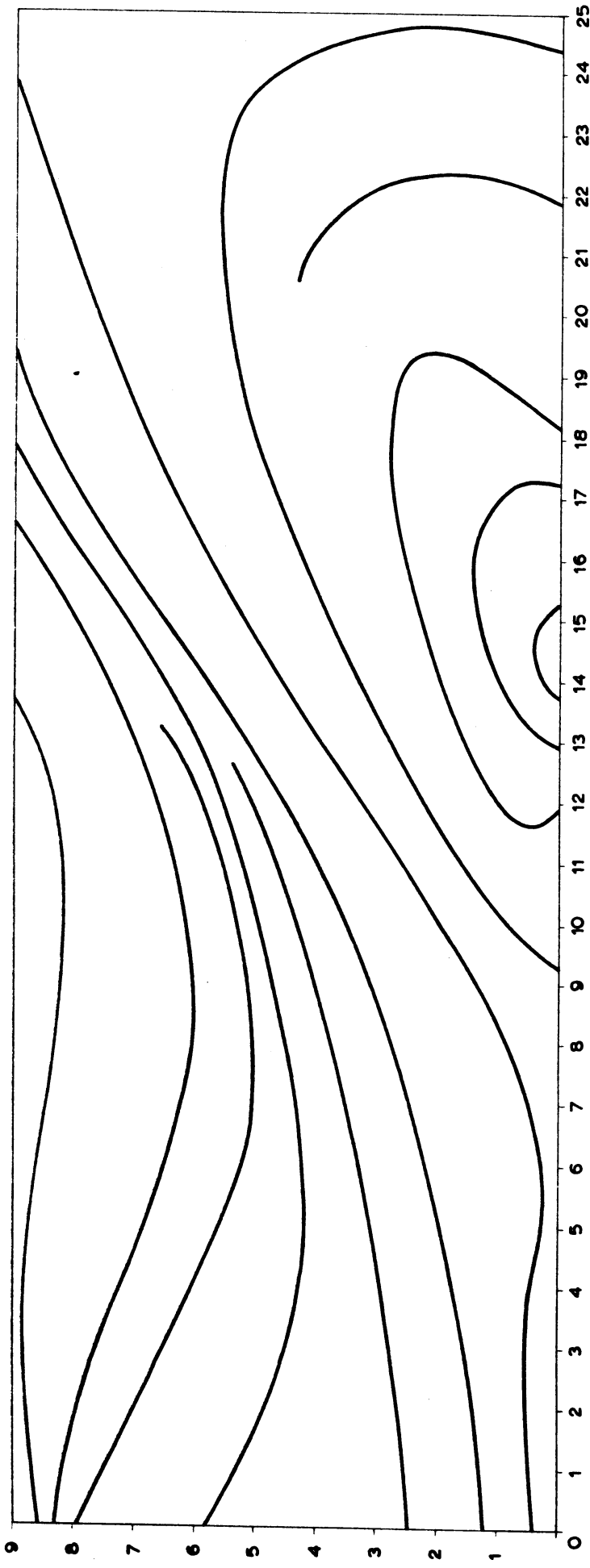


FIG 9 - II STROOMLIJNEN



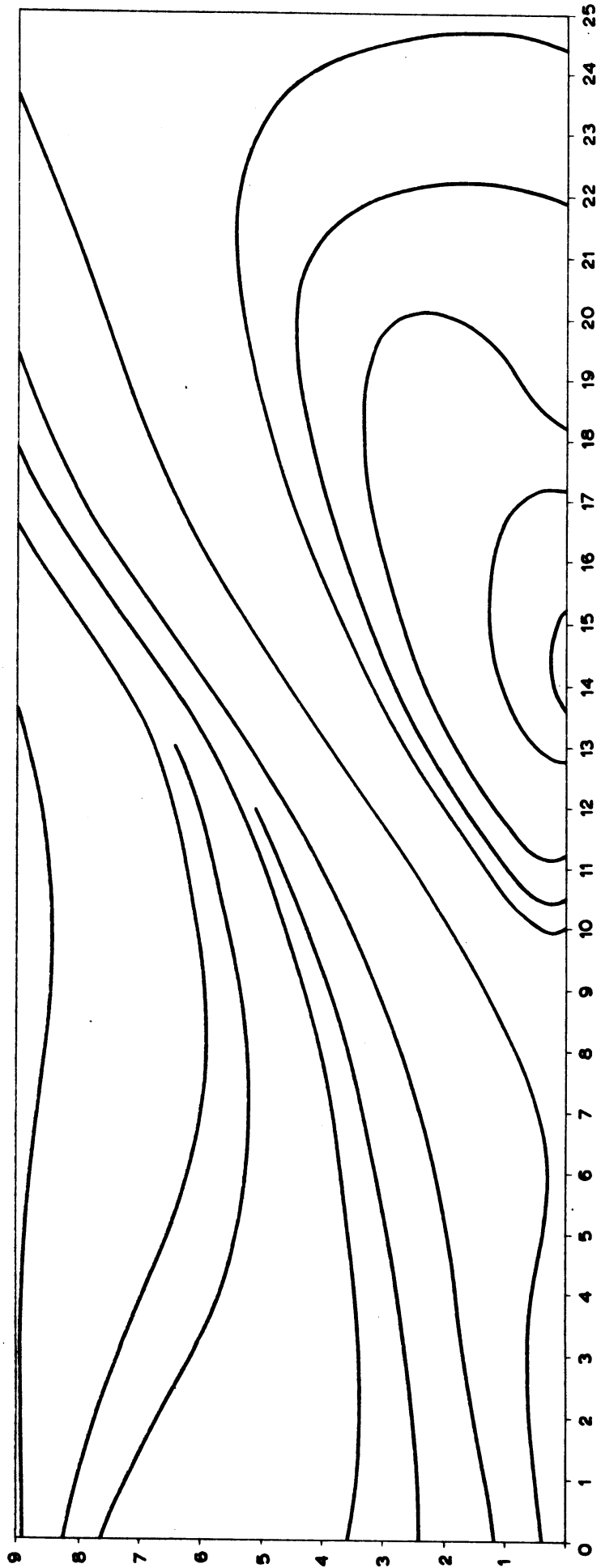


FIG.10-II STROOMLJNEN BEREKEND MET DE PICARD METHODE VOOR  $h_0 = 0.2$

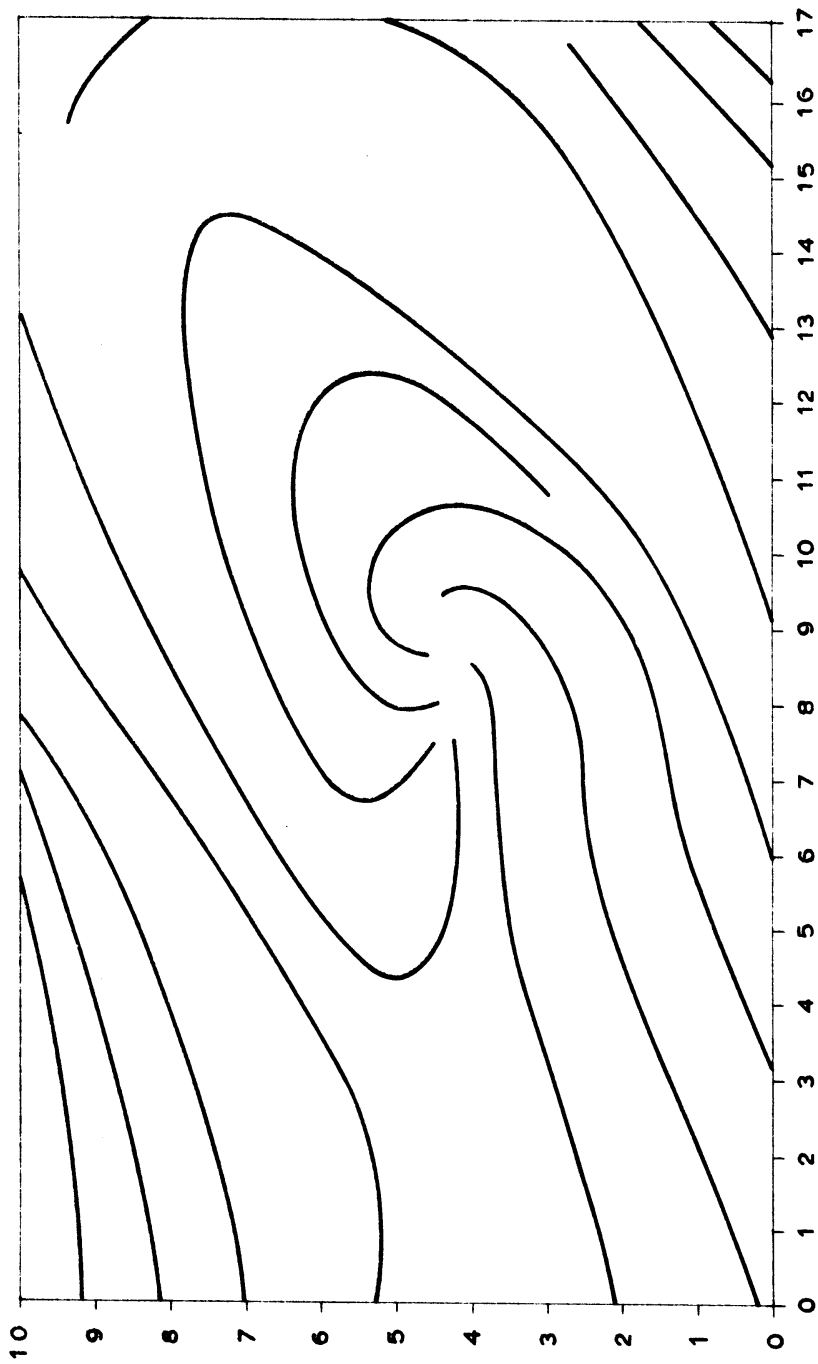


FIG. 11-III STROOMLJUNEN

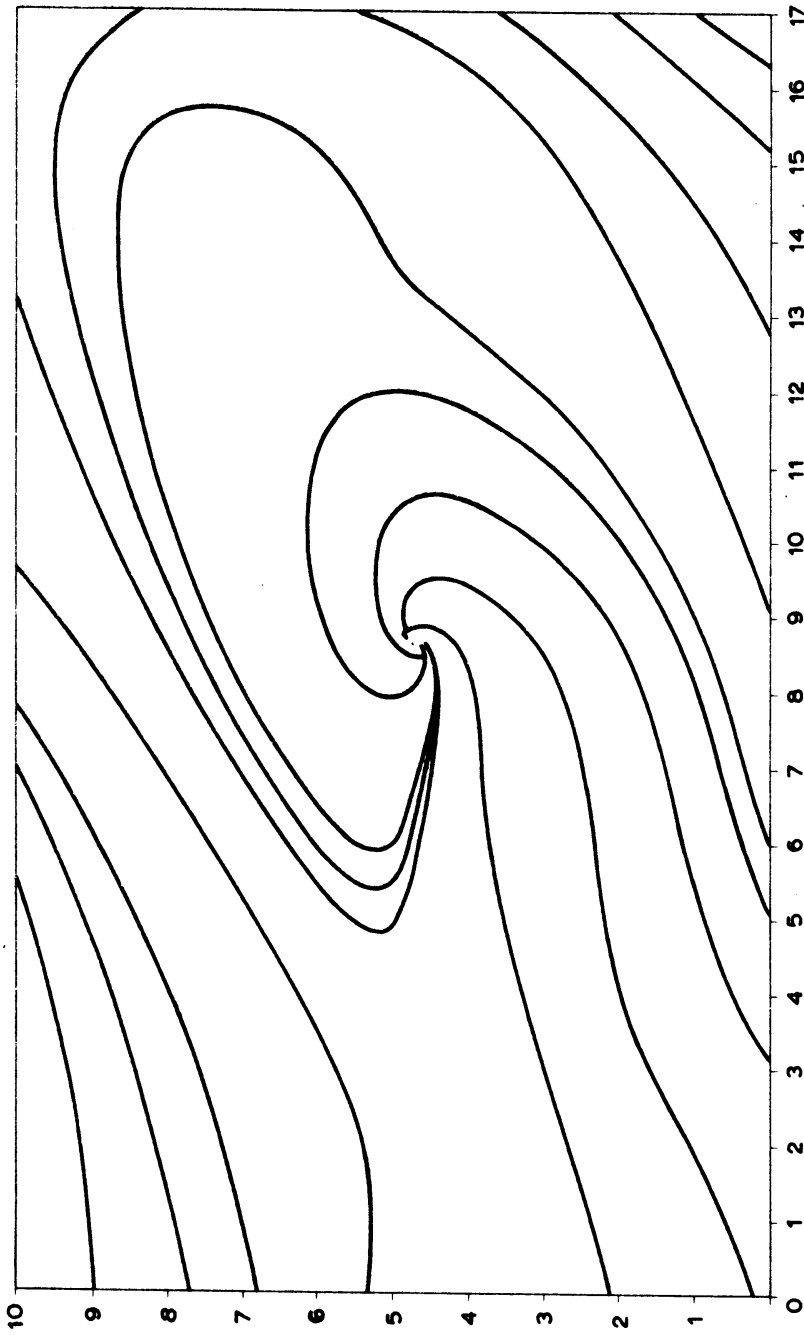


FIG. 12-III STROOMLJNEN BEREKEND MET DE PICARD METHODE VOOR  $h_0=0.2$

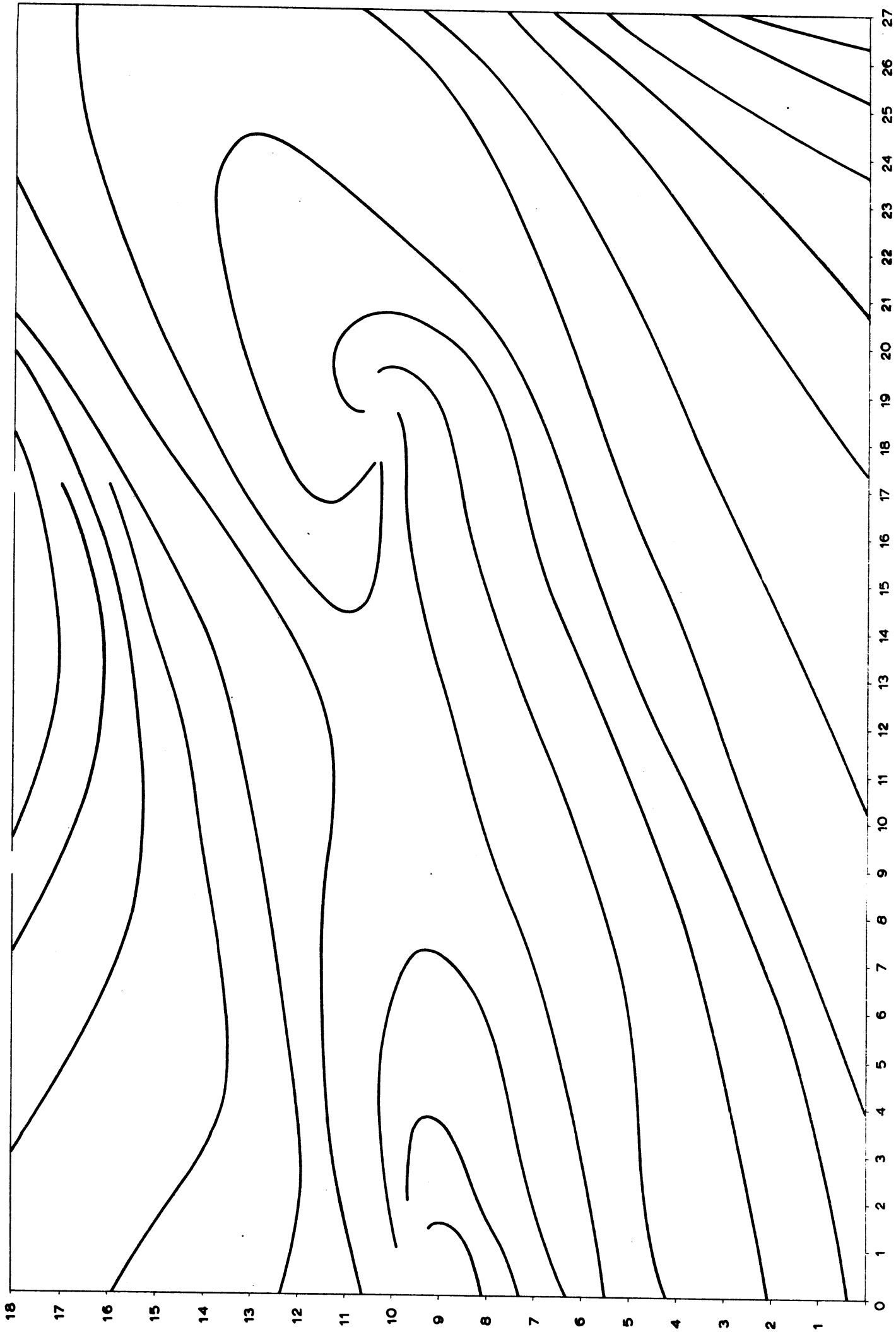


FIG. 13-IV STROOMLJNEN

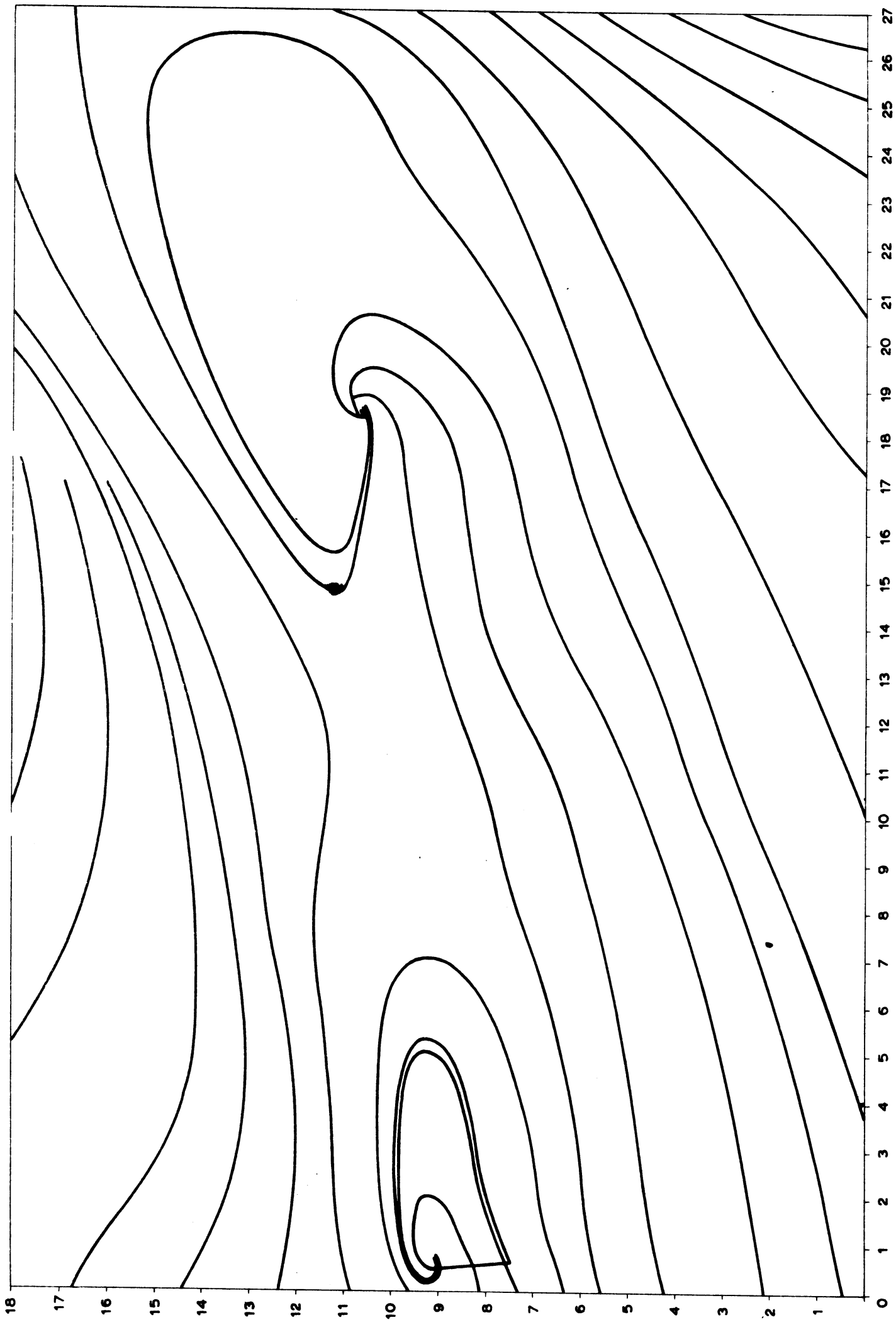


FIG.14-IV STROOMLIJNEN BEREKEND MET DE PICARD METHODE VOOR  $h_0=0.2$

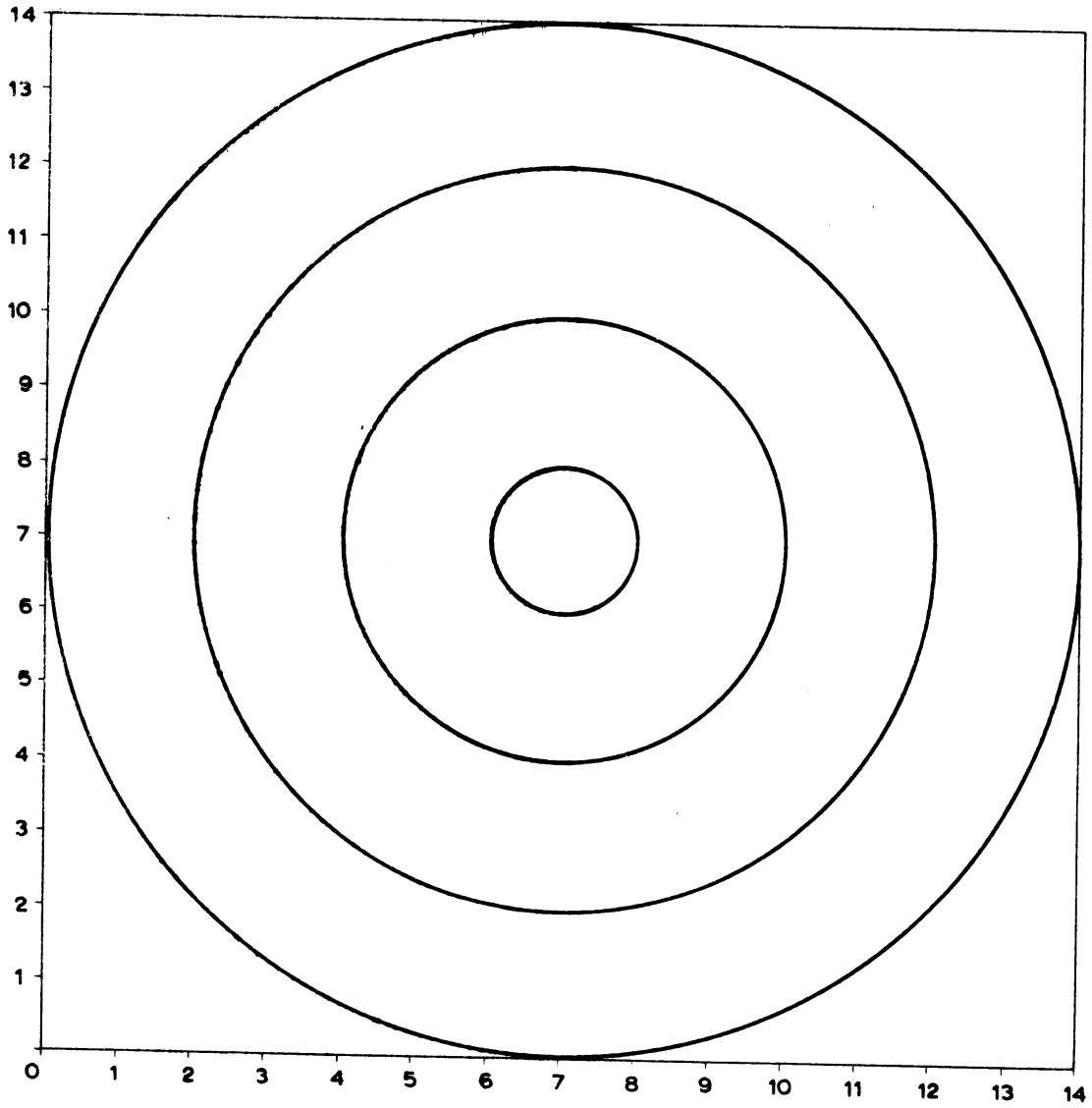


FIG. 15 - V

STROOMLIJEN - CIRKELS - BEREKEND MET DE PICARD METHODE VOOR  $h_0 = 0.2$